User Manual

for the Controllers **S56**, **S57**, **S59** and **S67**,

for the Display Unit **S83010** and for the

Configuration-Software **S57901**

SEAL

User Manual

for the Controllers **S56**, **S57**, **S59** and **S67**,

for the Display Unit **S83010** and for the

Configuration-Software **S57901**

SEAL

Important Notice

SEAL warrants that the product sold pursuant to any resultant agreement shall be free from defects in material or workmanship at the time of delivery. Such warranty shall extend for one (1) year, unless specified otherwise, beginning from the date of shipment. The buyer must provide notice to ***SEAL*** within this prescribed warranty period of any defect. If the defect is not the result of improper usage, service, maintenance or installation and the equipment has not been otherwise damaged or modified after delivery, ***SEAL*** shall either replace or repair the defective part or parts of equipment or replace the equipment. Shipment to ***SEAL***'s facility shall be borne on account of the buyer, including all duties, fees, brokerage and special handling charges. Shipment from ***SEAL***'s facility shall be borne by ***SEAL***.

SEAL PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. ***SEAL*** may revise this publication from time to time without notice.

SEAL SHALL NOT BE LIABLE FOR ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES INCURRED AS A RESULT OF ANY DEFECT IN ANY EQUIPMENT SOLD HEREUNDER AND ***SEAL***'s LIABILITY IS SPECIFICALLY LIMITED TO ITS OBLIGATION DESCRIBED HEREIN TO REPAIR OR REPLACE DEFECTIVE PARTS COVERED BY THIS WARRANTY.

Printed in Switzerland

DO-99.05-S57901-UM-E

SEAL AG
PO-Box 2103
CH-5430 Wettingen 2

Table of Content

List of Figures

List of Tables

CHAPTER 1 Introduction

This manual contains general information to make you familiar with the capabilities and operation of the ***SEAL S56, S57*** and ***S97*** devices. These are programmable industrial controllers with an expandable number of inputs and outputs. You can connect one or more of the devices ***S5701*** or ***S5703*** to one or more input/output devices from the ***SEAL S59xx*** series using the same field bus. The used field bus is the redundant twisted pair token bus, as specified in the IEEE 802.4 standard.

The controllers ***S5610, S5611, S5612*** and ***S5614*** support extension devices of the ***S56*** series with an RS485 field bus.

The use of this programmable controller with the software ***S57901*** requires no programming knowledge.

The display unit ***S83010*** Industrial Terminal may be used to control any of the master devices and is described in the last chapter. It may also be programmed by the software described in this manual.

1.1 Installation and Operation of S5701

The connector panel provides the connections for input/outputs, the twisted pair field bus and the power supply. The device features also two 25-pin RS232 connectors; the female connector is used for the connection to the PC.

1.1.1 Input/Output Connection

The connector panel shows how to connect inputs and outputs to the device.

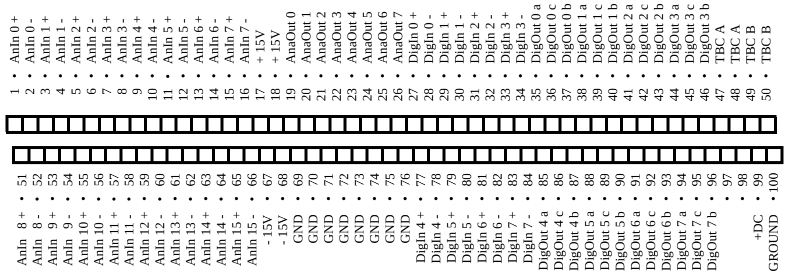


FIGURE 1 S5701: Connector Panel

The following table shows how to use the counter inputs:

	Channel A	Channel B	Reset (if Jumper set)
Counter 1	Dig In 0	Dig In 1	Dig In 4
Counter 2	Dig In 2	Dig In 3	Dig In 5

TABLE 1 S5701: Connection of Counters (Quadrature Decoder)

1.1.2 Field Bus

The field bus (TBC) connection is also done according to the connector panel. The redundant field bus capability is not supported in the current software version of the ***S5701*** device. The communication channel A has to be used. The field bus connection is polarity independent.

The lower 5 bits of the field bus address are set by the switch SW1. The address bit 0 is always zero. Switch 1.1 is address bit 1, switch 1.2 address bit 2 etc. If all address bits 00 - 05 are set to zero (all switches off), your S5701 device makes a hard reset every time after power on. The user program and buffer contents will be lost after power off in this case! If the field bus is not in use, these 5 switches must be set to 1 to avoid loss of user program and in the software the address of the device must be set to 1.

1.1.3 Dip Switch

Bits 7 and 8 are used to select the speed of the serial interfaces B (male) and A (female), respectively:

switch	0	1
SW1.7 (B)	9600 Baud	2400 Baud
SW1.8 (A)	9600 Baud	2400 Baud

TABLE 2 S5701: Baud Rates

1.1.4 Power Connection

The power supply (9-40V_{DC}) has to be connected to the connector panel. **It is very important to avoid false polarity, as this will cause damage to the controller.**

1.1.5 PC Connection

The female 25-pin connector has to be used for this purpose.

Pin	RS232	V.24	Direction	Description
1	AA	101	Both	Protective Ground
2	BA	103	To S5701	Receive Data
3	BB	104	From S5701	Transmit Data
4	CF	109	To S5701	CD - only for Modems (Modem Pin 8)
5	CD	108	From S5701	DTR - only for Modems (Pin 4 and 20)
6	CC	107	From S5701	470 Ohm pull up to 12V
7	AB	102	Both	Signal Ground
8	CF	109	From S5701	470 Ohm pull up to 12V

TABLE 3 S5701: Female RS232 Connector, Pin Description

For the connection to the PC, only pins 2, 3 and 7 are needed. Select the proper baud rate (usually 9600 Baud) according to Figure 9 on page 1-9. To connect a modem, pin 4 has to be connected to pin 8 of the modem (CD) and pin 5 has to be connected to modem pins 4 and 20 (RTS and DTR).

The male 25-pin connector is used to connect extension modules S59xxE:

Pin	RS232	V.24	Direction	Description
1	AA	101	Both	Protective Ground
2	BA	103	From S5701	Transmit Data
3	BB	104	To S5701	Receive Data
7	AB	102	Both	Signal Ground

TABLE 4 S5701: Male RS232 Connector, Pin Description

1.1.6 Operating

1.1.6.1 LED Indicators

The LEDs indicate the state of the ***SEAL S5701*** device.

l green	device is running	l green	not used
l green	user program is active	l green	not used
		l green	digital output 7
l yellow	communication to PC or S83010	l green	digital output 6
l yellow	communication to field bus	l green	digital output 5
		l green	digital output 4
l red	watch dog active (error)	l green	digital output 3
l red	hardware failure	l green	digital output 2
		l green	digital output 1
		l green	digital output 0

FIGURE 2 S5701: LED Indicators

1.1.6.2 Reset

If any problems with the ***SEAL S5701*** device arise, a “hard reset” can be made. After a hard reset all memory (buffers, user program) will be completely cleared.

To “hard reset” the device set all the address switches to zero before turning on the power.

1.1.6.3 Address and Baud Rates

The field bus address and the baud rates for the serial communications ports have to be selected by the dip switch.

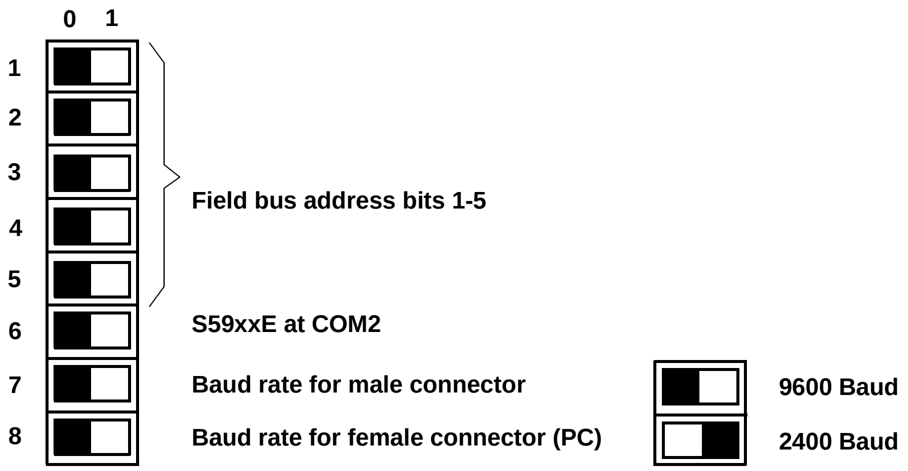

0
1
1
2
3
4
5
6
7
8
Field bus address bits 1-5
S59xxE at COM2
Baud rate for male connector
Baud rate for female connector (PC)
9600 Baud
2400 Baud

FIGURE 3 S5701: Dip Switch Usage

1.2 Installation and Operation of S5703

Two connectors provide the connections for input/outputs. The twisted pair field bus and the power supply are connected to separate connectors. The device features also two 25-pin RS232 connectors; both connectors may be used for the connection to the PC, but only the female connector allows to load additional firmware. Modem features are supported by the female connector while the extension modules ***S59xxE*** may be connected to the male connector.

1.2.1 Input/Output Connection

The connector panel shows how to connect inputs and outputs to the device.

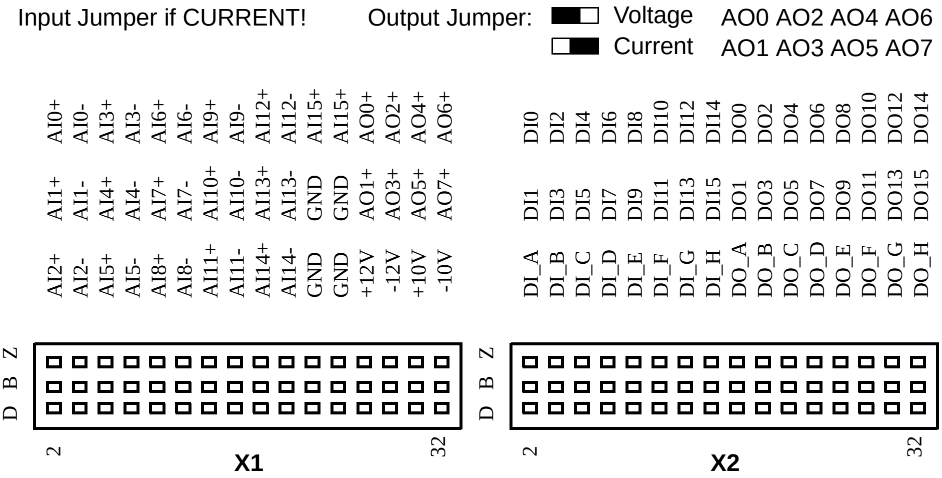

Input Jumper if CURRENT!
Output Jumper:
Voltage
Current
AO0 AO2 AO4 AO6
AO1 AO3 AO5 AO7
AI0+ AI0- AI3+ AI3- AI6+ AI6- AI9+ AI9- AI12+ AI12- AI15+ AI15+ AO0+ AO2+ AO4+ AO6+
AI1+ AI1- AI4+ AI4- AI7+ AI7- AI10+ AI10- AI13+ AI13- GND GND AO1+ AO3+ AO5+ AO7+
AI2+ AI2- AI5+ AI5- AI8+ AI8- AI11+ AI11- AI14+ AI14- GND GND +12V -12V +10V -10V
DI0 DI2 DI4 DI6 DI8 DI10 DI12 DI14 DO0 DO2 DO4 DO6 DO8 DO10 DO12 DO14
DI1 DI3 DI5 DI7 DI9 DI11 DI13 DI15 DO1 DO3 DO5 DO7 DO9 DO11 DO13 DO15
DI_A DI_B DI_C DI_D DI_E DI_F DI_G DI_H DO_A DO_B DO_C DO_D DO_E DO_F DO_G DO_H
D B Z
2
32
X1
D B Z
2
32
X2

FIGURE 4 S5703: Connector Panel, Inputs and Outputs

Common Input	Inputs	Common Output	Outputs
DI_A	DI0, DI1	DO_A	DO0, DO1
DI_B	DI2, DI3	DO_B	DO2, DO3
DI_C	DI4	DO_C	DO4, DO5
DI_D	DI5, DI6, DI7	DO_D	DO6, DO7
DI_E	DI8, DI9	DO_E	DO8, DO9
DI_F	DI10, DI11	DO_F	DO10, DO11
DI_G	DI12, DI13	DO_G	DO12, DO13
DI_H	DI14, DI15	DO_H	DO14, DO15

TABLE 5 S5703: Digital I/O Connections

The following table shows the usage of the counter inputs:

Pulses (n)	Velocity (n/s)	Channel A	Channel B	Reset (Jumper)	Type of Counter
Counter 0	Counter 2	Dig In 0	Dig In 1	Dig In 4	Incremental
Counter 1	Counter 3	Dig In 2	Dig In 3	Dig In 5	Incremental
Counter 4	Counter 8	Dig In 6			Software
Counter 5	Counter 9	Dig In 7			Software
Counter 6	Counter 10	Dig In 8			Software
Counter 7	Counter 11	Dig In 9			Software
Counter 12		Dig In 10			Software
Counter 13		Dig In 11			Software
Counter 14		Dig In 12			Software
Counter 15		Dig In 13			Software

TABLE 6 S5703: Counter Inputs

The incremental decoder inputs may count up to 1 MHz. The software counters may count up to half the cycle frequency (500 Hz at 1 ms). If the counters 0/2 or 1/3 are selected as pulse counters each edge will be counted twice; counters 4 to 15 count only rising edges.

Below the wiring of the digital I/Os is shown:

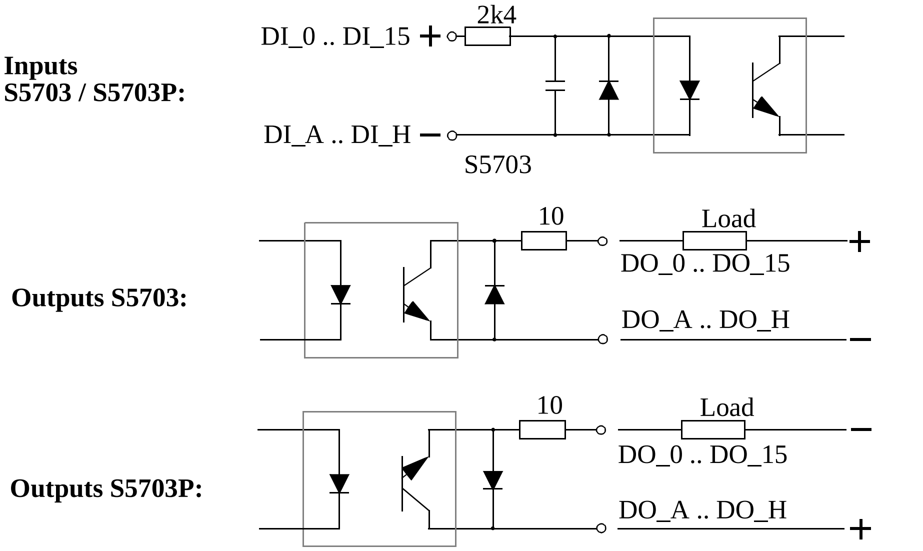


FIGURE 5 S5703: Digital Inputs and Outputs

1.2.2 Field Bus

The field bus (TBC) connection is done using the 5-pin connector. The redundant field bus capability is not supported in the current software version of the ***S5703*** device. The communication channel A has to be used. The field bus connection is polarity independent.

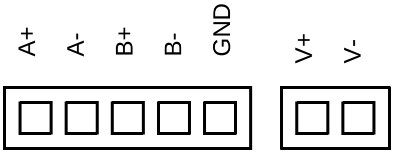


FIGURE 6 S5703: Connector Panel, Field Bus and Power

The field bus address is set by the switch SW1. The address bit 0 is always zero. Switch 1.8 is address bit 1, switch 1.7 address bit 2 etc. **If the fieldbus is not used, all bits of switch SW1 have to be set to 1 an in the software address 1 has to be selected.**

1.2.3 PC Connection

The female 25-pin connector has to be used for this purpose.

Pin	RS232	V.24	Direction	Description
1	AA	101	Both	Protective Ground
2	BA	103	To S5703	Receive Data
3	BB	104	From S5703	Transmit Data
4	CF	109	To S5703	CD - only for Modems (Modem Pin 8)
5	CD	108	From S5703	DTR - only for Modems (Pin 4 and 20)
6	CC	107	From S5703	470 Ohm pull up to 12V
7	AB	102	Both	Signal Ground
8	CF	109	From S5703	470 Ohm pull up to 12V

TABLE 7 S5703: Female RS232 Connector, Pin Description

For the connection to the PC, only pins 2, 3 and 7 are needed. To connect a modem, pin 4 has to be connected to pin 8 of the modem (CD) and pin 5 has to be connected to modem pins 4 and 20 (RTS and DTR). Select the proper baud rate (usually 9600 Baud) according to Figure 9 on page 1-9. The male 25-pin connector is used to connect extension modules ***S59xxE***:

Pin	RS232	V.24	Direction	Description
1	AA	101	Both	Protective Ground
2	BA	103	From S5701	Transmit Data
3	BB	104	To S5701	Receive Data
7	AB	102	Both	Signal Ground

TABLE 8 S5703: Male RS232 Connector, Pin Description

1.2.4 Power Connection

The power supply (15-35V$_{DC}$) has to be connected to the 2-pin connector.

1.2.5 Position on the Field Bus

Each ***SEAL S5703*** or ***SEAL S59xx*** device can take any position on the field bus.

1.2.6 Operating

1.2.6.1 LED Indicators

The LEDs indicate the state of the SEAL S5703 device.

l green	device is running	l green	input
l green	user program is active	l green	output
		l yellow	I/O indicator 0
l yellow	communication to PC or S83010	l yellow	I/O indicator 1
l yellow	communication to field bus	l yellow	I/O indicator 2
		l yellow	I/O indicator 3
l red	firmware checksum error	l green	cycle violation 0
l red	no firmware loaded	l green	cycle violation 1
		l green	cycle violation 2
		l green	cycle violation 3

FIGURE 7 S5703: LED Indicators

The I/O indicator LEDs show the address of the currently displayed input and output in the binary format from 0 to 15 with indicator 0 as the LSB and indicator 3 as the MSB.

The cycle violation LEDs display the number of additional cycles necessary to calculate the user software (if any).

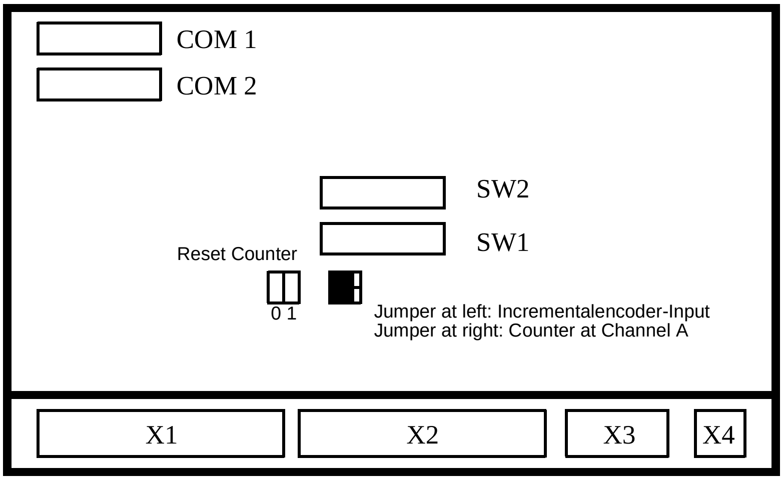

COM 1
COM 2
SW2
SW1
Reset Counter
0 1
Jumper at left: Incrementalencoder-Input
Jumper at right: Counter at Channel A
X1
X2
X3
X4

FIGURE 8 S5703: Connectors and Switches

1.2.6.2 Dipswitches

The upper dipswitches have to be used to select the baud rates for the serial communications ports, to disable the loaded firmware and to select the cycle time.

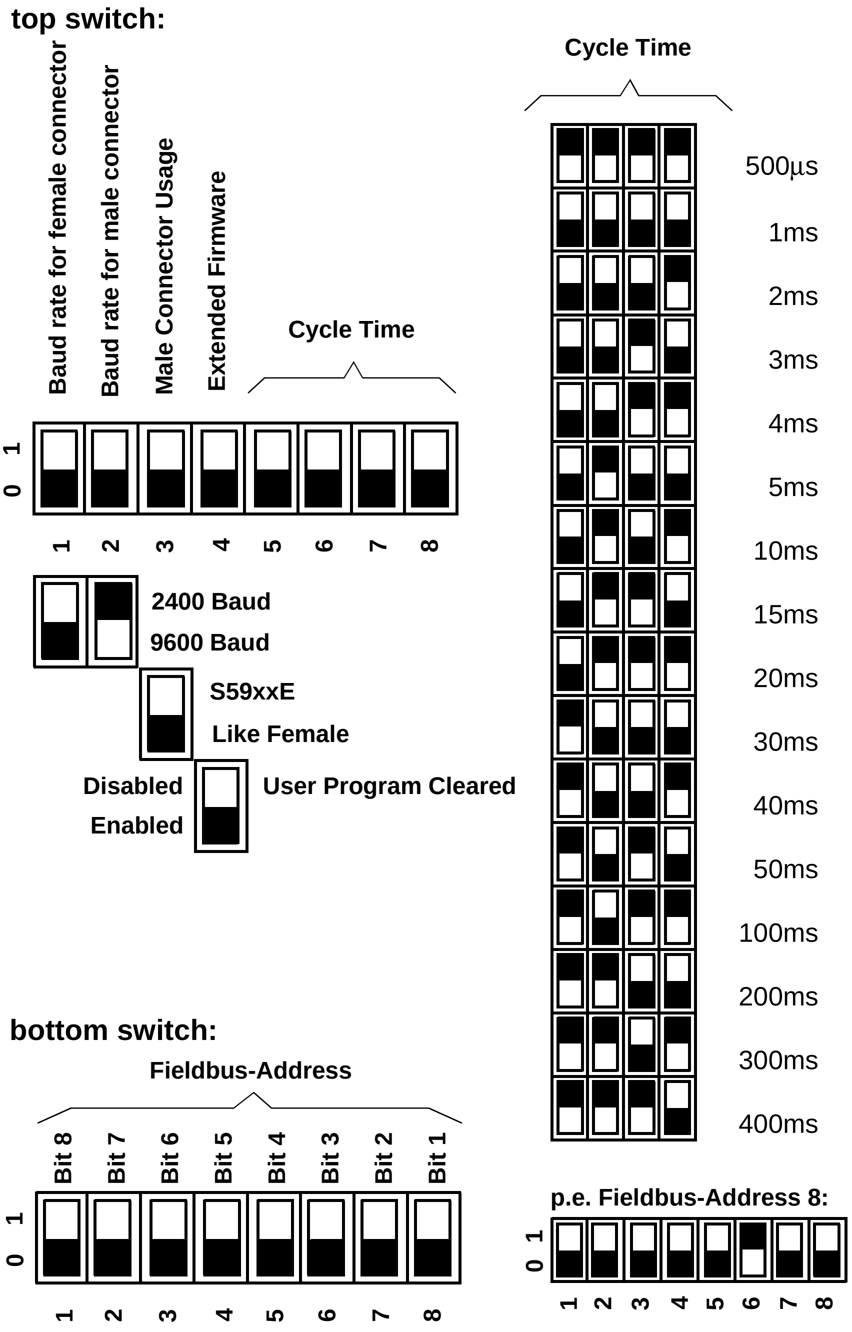


FIGURE 9 S5703: Dip Switch Usage

The field bus address has to be selected by the second dipswitch. If no field bus is installed, this switch must be entirely switched to 0 or 1, selecting address 0 or 255.

1.3 Installation and Operation of S5611C

All inputs and outputs, the power supply and the field bus are connected to a 48-pol. DIN connector.

1.3.1 Input- and Output Connections

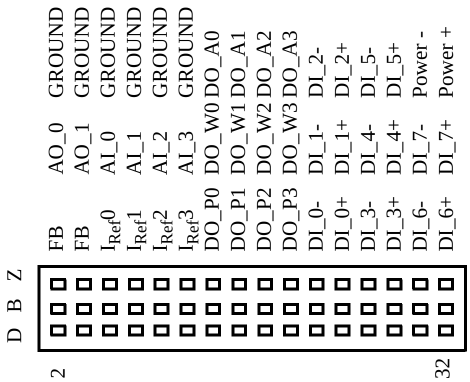


FIGURE 10 S5611C: DIN Connector

Pin	Description
FB	Fieldbus, polarity independent
AO	Analog Output
GROUND	Ground
AI	Analog Input
I_{Ref}	Current Reference (Pt_{100} or Pt_{1000})
DO_A	Digital Output, active
DO_W	Digital Output, root
DO_P	Digital Output, passive
DI+, DI-	Digital Input, galvanically insulated
Power	Supply Voltage, 12 - 24V_{DC}

TABLE 9 S5611C: Description of the Connector Pins

1.3.2 Field Bus

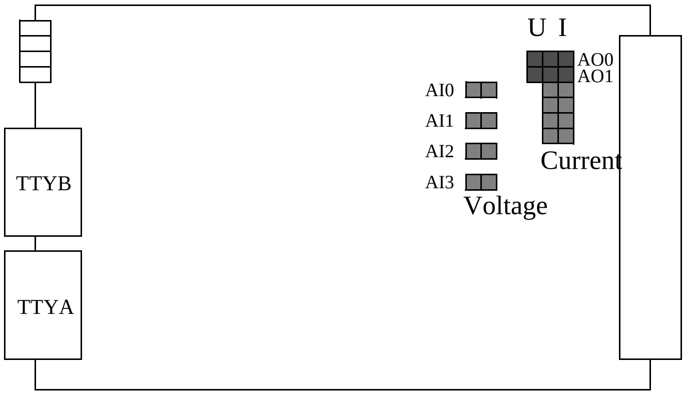
The field bus has to be connected as described above. The polarity of the cable does not matter. The address of ***S5611C*** is always 1.

1.3.3 Power Supply

The supply voltage (12-24V_{DC}) has to be connected to the 48-Pol. connector, too.

1.3.4 Configuration

Inside of the device there are several jumpers to adjust inputs and outputs:



■ Analog Outputs: at left voltage, at right current

▨ Analog Inputs: at left voltage, at right current,none for resistance (Pt100)

FIGURE 11 S5611C: Configuration of Inputs and Outputs

1.3.5 Operation

1.3.5.1 LED Display

The LEDs show the current state of a ***SEAL S5611C***.

l green	Device has power
l green	User program: short pauses = active; long pauses = inactive
l yellow	Communication to PC or S83010
l yellow	Communication using the Field-Bus

FIGURE 12 S5611C: LED Display

1.3.5.2 Cycle Time

Selecting a cycle time from 30 to 400 ms allows to use up to 64 I/O-addresses (32 external modules). For fast applications it is possible to select 5 to 20 ms, then only 12 I/O-adresses are supported (up to 6 external devices).

1.3.6 Serial Interfaces

The female DSUB connector (TTYB) is used to connect the PC.

Pin	RS232	V.24	Direction	Description
2	BB	104	from S5611C	Transmit Data
3	BA	103	to S5611C	Receive Data
5	AB	102	both	Signal Ground
6	CC	107	from S5611C	470 Ohm pull up to 12V

TABLE 10 S5611C: female RS232 Connector, Pin Description

The male connector (TTYA) is used to connect a modem or a ***SEAL SLink21*** ISDN-Terminaladaper or the display unit ***S83010*** using a standard 1:1 cable. A PC may also be connected using a null-modem-cable:

Pin	RS232	V.24	Direction	Description
1	CF	109	to S5611C	CD - Carrier Detect
2	BA	103	to S5611C	Receive Data
3	BB	104	from S5611C	Transmit Data
4	CD	108	from S5611C	DTR - Data Terminal Ready
5	AB	102	both	Signal Ground
6			to S5611C	DSR
7			from S5611C	RTS
8			to S5611C	CTS
9			to S5611C	RING

TABLE 11 S5611C: male RS232 Connector, Pin Description

DTR is always active to allow incoming calls. Before and after sending a message DTR is removed to ensure modem hangup. A connection is recognized if CD is active.

If CD remains activ (also if not connected), a failed hangup is indicated at the alarm function block and the message is sent without dialing.

1.4 Installation and Operation of S5611D with integrated Display

The controller S5611D combines the features of S5611C with the display unut S83010. All inputs and outputs are connected to the 48-pol. DIN connector.

1.4.1 Input- and Output Connections

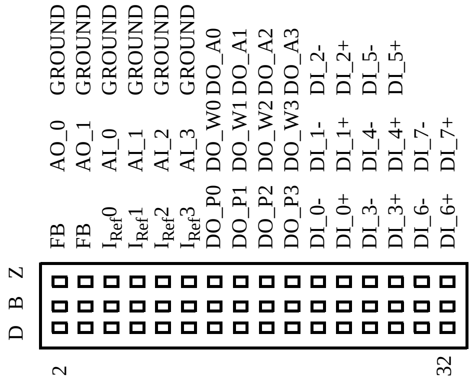


FIGURE 13 S5611D: DIN Connector

Pin	Description
FB	Fieldbus, polarity independent
AO	Analog Output
GROUND	Ground
AI	Analog Input
I_{Ref}	Current Reference (Pt_{100} or Pt_{1000})
DO_A	Digital Output, active
DO_W	Digital Output, root
DO_P	Digital Output, passive
DI+, DI-	Digital Input, galvanically insulated

TABLE 12 S5611D: Description of the Connector Pins

1.4.2 Field Bus

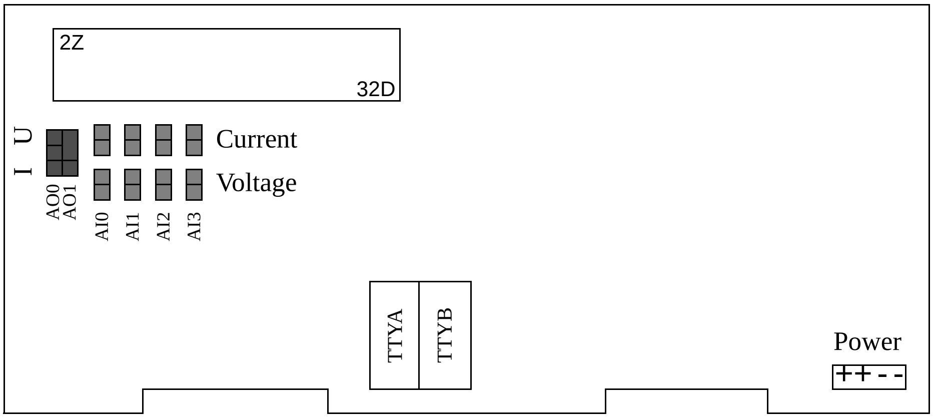
The field bus has to be connected as described above. The polarity of the cable does not matter. The address of ***S5611D*** is always 1.

1.4.3 Power Supply

The supply voltage (12-24V_{DC}) has to be connected to the small 4-pol. connector.

1.4.4 Configuration

Inside of the device there are several jumpers to adjust inputs and outputs:



■ Analog Outputs: at top current, at bottom voltage

▨ Analog Inputs: at top voltage, at bottom current,none for resistance (Pt100)

FIGURE 14 S5611D: Configuration of Inputs and Outputs, Connectors

1.4.5 Operation

1.4.5.1 LED Display

The LEDs show the current state of a ***SEAL S5611D***.

l red	Error
l yellow	Communication to PC
l green	Device running

FIGURE 15 S5611D: LED Display

The red and the green LED may be turned on or off by the user program.

1.4.5.2 Cycle Time

Selecting a cycle time from 30 to 400 ms allows to use up to 64 I/O-addresses (32 external modules). For fast applications it is possible to select 5 to 20 ms, then only 12 I/O-adresses are supported (up to 6 external devices).

1.4.5.3 Display

The programming of the display unit is done in the same way as with ***S83010***.

1.4.6 Serial Interfaces

The female DSUB connector (TTYB) is used to connect the PC.

Pin	RS232	V.24	Direction	Description
2	BB	104	from S5611D	Transmit Data
3	BA	103	to S5611D	Receive Data
5	AB	102	both	Signal Ground
6	CC	107	from S5611D	470 Ohm pull up to 12V

TABLE 13 S5611D: female RS232 Connector, Pin Description

The male connector (TTYA) is used to connect a modem or a ***SEAL SLink21*** ISDN-Terminaladaper using a standard 1:1 cable. A PC may also be connected using a null-modem-cable:

Pin	RS232	V.24	Direction	Description
1	CF	109	to S5611D	CD - Carrier Detect
2	BA	103	to S5611D	Receive Data
3	BB	104	from S5611D	Transmit Data
4	CD	108	from S5611D	DTR - Data Terminal Ready
5	AB	102	both	Signal Ground
6			to S5611D	DSR
7			from S5611D	RTS
8			to S5611D	CTS
9			to S5611D	RING

TABLE 14 S5611D: male RS232 Connector, Pin Description

DTR is always active to allow incoming calls. Before and after sending a message DTR is removed to ensure modem hangup. A connection is recognized if CD is active.

If CD remains activ (also if not connected), a failed hangup is indicated at the alarm function block and the message is sent without dialing.

1.5 Installation and Operation of S5612

S5612 is a powerful controller with built in ISDN. ***S5612*** is able to communicate to other ISDN controllers very fast. It is possible to setup ***S5612*** using ISDN as well as one of the serial interfaces. It is also possible to send messages to an ISDN Terminal adaptor (***SEAL SLink21***); for these messages or for configuration of ***S5612*** the protocol X.75 is used on the B channel.

1.5.1 Input- and Output Connections

The figure below shows the connectors of an ***S5612*** controller.

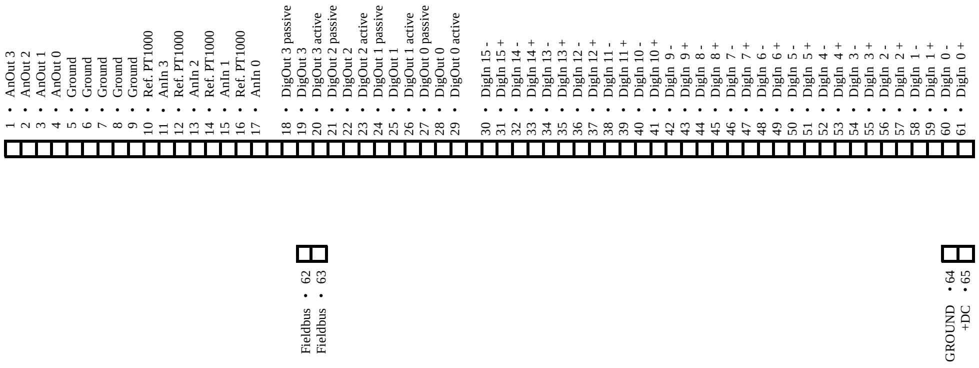


FIGURE 16 S5612: Connectors, Inputs, Outputs, Field Bus, Power

1.5.2 Field Bus

The fieldbus is polarity independent and the same as for ***S5610***.

1.5.3 Power Supply

The power (15-30V_{DC}) has to be connected as described in the figure above.

1.5.4 ISDN Connection

One of the RJ45 connectors has to be used.

1.5.5 Usage

The system LEDs (8) show the state of the ***SEAL S5612*** controller.

1.5.6 Configuration

Several jumpers within the controller are used to adjust inputs and outputs of the controller:

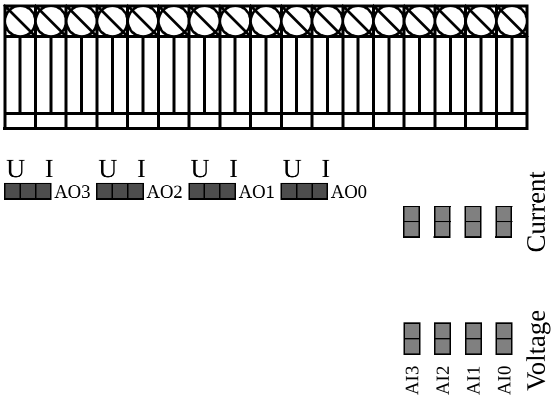


FIGURE 17 S5612: Jumper Configuration

For Pt1000 measurement no jumpers have to be inserted.

1.5.7 PC Connection

One of the 9-Pol. female DSUB connectors is used:

Pin	RS232	V.24	Direction	Description
2	BB	104	from S5612	sent data
3	BA	103	to S5612	received data
5	AB	102	both	ground

TABLE 15 S5612: RS232 Connectors

Only pins 2, 3 and 5 are used for the connection to the PC. The baud rate is 9600 and may be changed if desired.

1.5.8 Software Configuration

The chapter about the ***S9704*** ISDN Controller contains some hints for the ISDN Configuration.

1.6 Installation and Operation of S5614

S5614 is a powerful controller with built in ISDN and Ethernet as well as an alphameric LED Display containing 16 digits. ***S5614*** is able to communicate to other ISDN controllers very fast. It is possible to setup ***S5614*** using ISDN as well as the serial interface. It is also possible to send messages to an ISDN Terminal adaptor (***SEAL SLink21***); for these messages or for configuration of ***S5614*** the protocol X.75 is used on the B channel.

1.6.1 Input- and Output Connections

The figure below shows the connectors of an ***S5614*** controller.

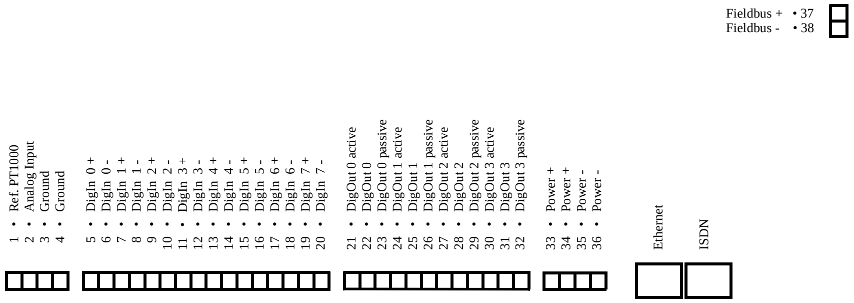


FIGURE 18 S5614: Connectors, Inputs, Outputs, Field Bus, Power

1.6.2 Field Bus

The fieldbus is polarity independent and the same as for ***S5612***.

1.6.3 Power Supply

The power (15-30V_{DC}) has to be connected as described in the figure above.

1.6.4 ISDN Connection

The RJ45 connector on the right hand side has to be used.

1.6.5 Ethernet

The RJ45 connector on the left hand side has to be used.

1.6.6 Configuration

Several jumpers within the controller are used to adjust inputs and outputs of the controller:

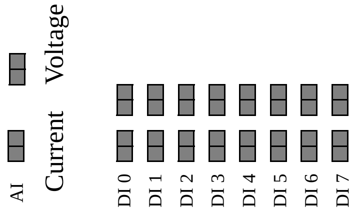


FIGURE 19 S5614: Jumper Configuration

For Pt1000 measurement no jumpers have to be inserted.

If an external voltage is connected to the digital inputs, no jumpers have to be inserted. If an external switch like a relay is used, both he upper and the lower jumper has to be inserted.

1.6.7 PC Connection

The 9-Pol. female DSUB connector is used:

Pin	RS232	V.24	Direction	Description
2	BB	104	from S5614	sent data
3	BA	103	to S5614	received data
5	AB	102	both	ground

TABLE 16 S5614: RS232 Connectors

Only pins 2, 3 and 5 are used for the connection to the PC. The baud rate is 9600 and may be changed by the PC software if desired.

1.6.8 Software Configuration

The chapter about the ***S9704*** ISDN controller contains some hints for ISDN Configuration.

Using the function block “Message” it is possible to send messages to the built in display.

1.7 Installation and Operation of S5651

1.7.1 Inputs and Outputs

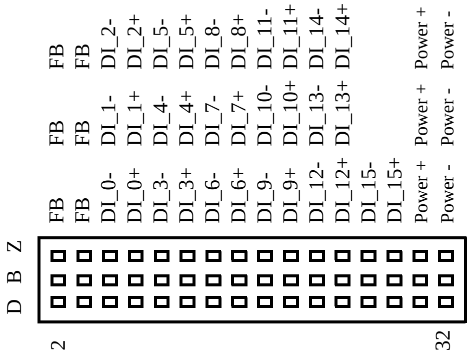


FIGURE 20 S5651: Input- and Output Connector

Pin	Description
FB	Fieldbus, polarity independent
DI+, DI-	Digital Input, galvanically insulated
Power	Power Supply, 9-30V_{DC}

TABLE 17 S5651: Description of Connector Pins

1.7.2 Power Supply

The supply voltage (9-30V_{DC}) has to be connected to the DIN connector.

1.7.3 Operation

The LED indicates the operation of the ***S5651***. Fast blinking shows that there is no connection to an ***S5610*** master. Slow blinking shows that ***S5651*** sends data to ***S5610.***

1.7.4 Field Bus

The field bus address of the ***S5651*** has to be selected using the five jumpers to select a value from 6 to 31:

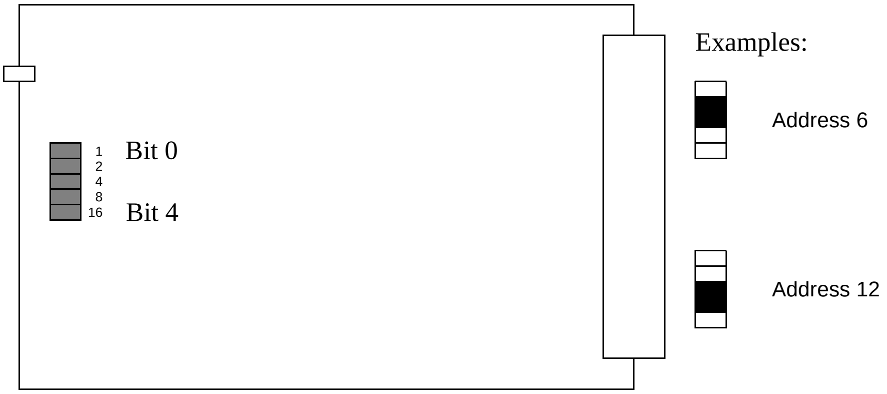


FIGURE 21 S5651: Address Selection

1.7.5 Configuration of the Digital Inputs

If an external voltage is connected to the digital inputs, no jumpers have to be inserted. If an external switch like a relay is used, both the left and the right jumper has to be inserted:

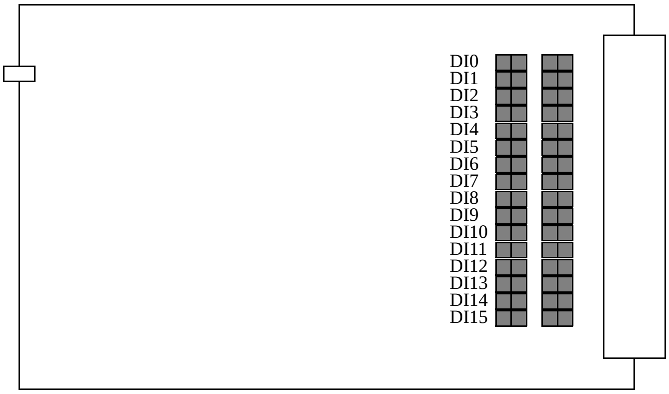


FIGURE 22 S5651: Configuration of Digital Inputs

1.8 Installation and Operation of S5661

1.8.1 Input- and Output Connections

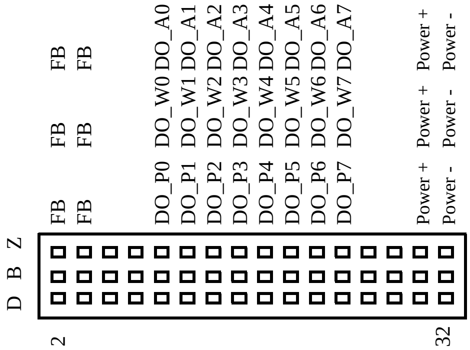


FIGURE 23 S5661: Connector

Pin	Description
FB	Field bus, polarity independent
DO_P	Digital Output, passive
DO_W	Digital Output, root
DO_A	Digital Output, active
Power	Power Supply, 9-30V_{DC}

TABLE 18 S5661: Pin Description

1.8.2 Field Bus

The field bus address of the ***S5661*** has to be selected using the five jumpers to select a value from 6 to 31:

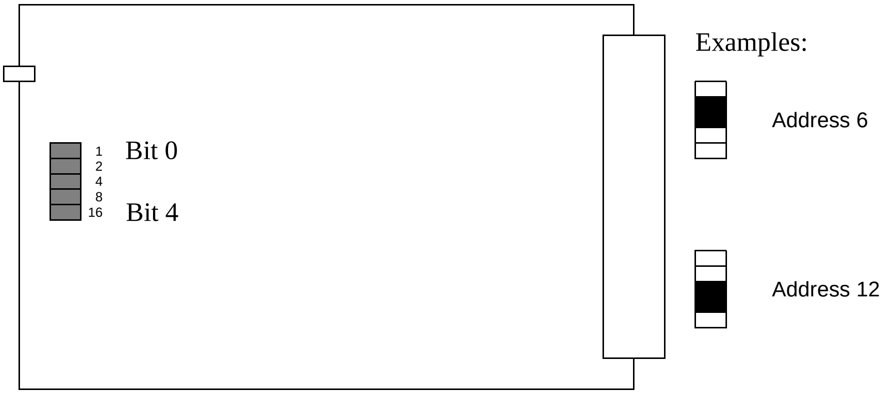


FIGURE 24 S5661: Address Selection

1.8.3 Power Supply

The supply voltage (9-30V_{DC}) has to be connected to the DIN connector.

1.8.4 Operation

The LED indicates the operation of the ***S5661***. Fast blinking shows that there is no connection to an ***S5610*** master. Slow blinking shows that ***S5661*** receives data from ***S5610.*** If ***S5661*** lost the connection to ***S5610*** the LED blinks a little bit faster. After 5 minutes all relays are turned off and the LED of ***S5661*** blinks fast.

1.9 Installation and Operation of S5671

1.9.1 Input- and Output Connections

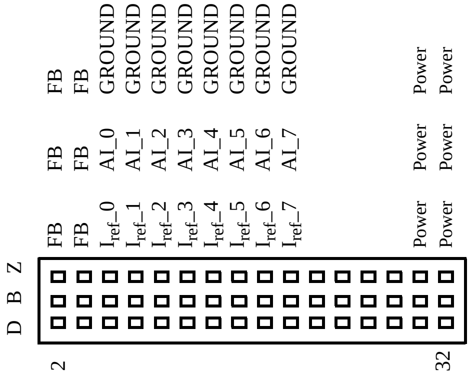


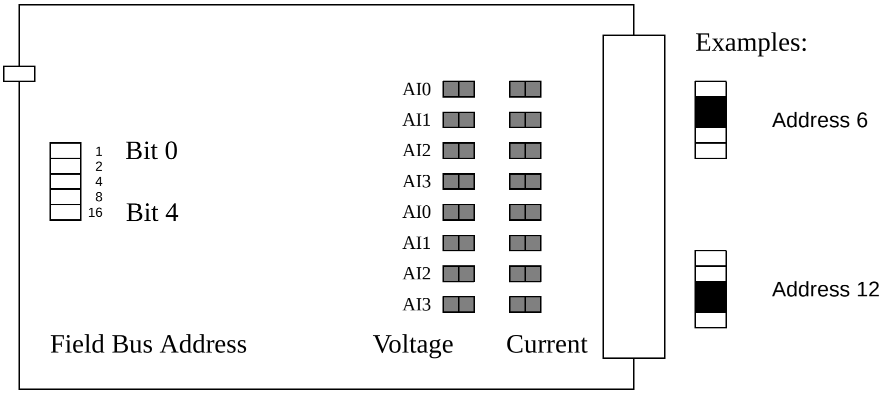
FIGURE 25 S5671: Connector

Pin	Description
FB	Field bus, polarity independent
GROUND	Ground
AI	Analog Input
I_{Ref}	Current Reference (Pt_{100})
Power	Power Supply, 9-12V_{AC} or 9-12V_{DC}

TABLE 19 S5671: Pin Description

1.9.2 Field Bus, Configuration

The field bus address of ***S5671*** is selected by the 5 jumpers like ***S5651*** or ***S5661***. Additionally the inputs have to be selected according to the desired functionality.



▨ Analog Inputs: at left voltage, at right current, none for resistance (Pt100)

FIGURE 26 S5671: Address- and Input Configuration

1.9.3 Power Supply

The supply voltage (9-12$V_{AC/DC}$) has to be connected to the DIN connector.

1.9.4 Operation

The LED indicates the operation of the ***S5671***. Fast blinking shows that there is no connection to an ***S5610*** master. Slow blinking shows that ***S5671*** sends data to ***S5610.***

1.10 Installation and Operation of S5681

1.10.1 Input- and Output Connections

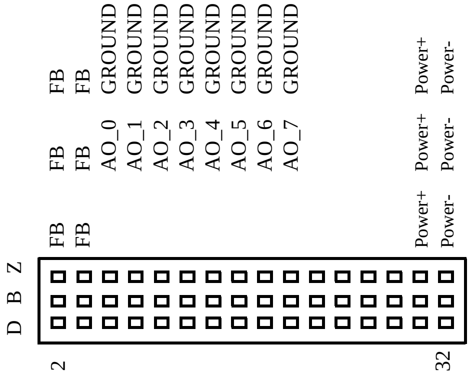


FIGURE 27 S5681: Connector

Pin	Description
FB	Field bus, polarity independent
GROUND	Ground
AO	Analog Output
Power	Power Supply, 9-30V_{DC}

TABLE 20 S5681: Pin Description

1.10.2 Field Bus, Configuration

The field bus address of ***S5681*** is selected by the 5 jumpers like ***S5651*** or ***S5661***. Additionally the inputs have to be selected according to the desired functionality.

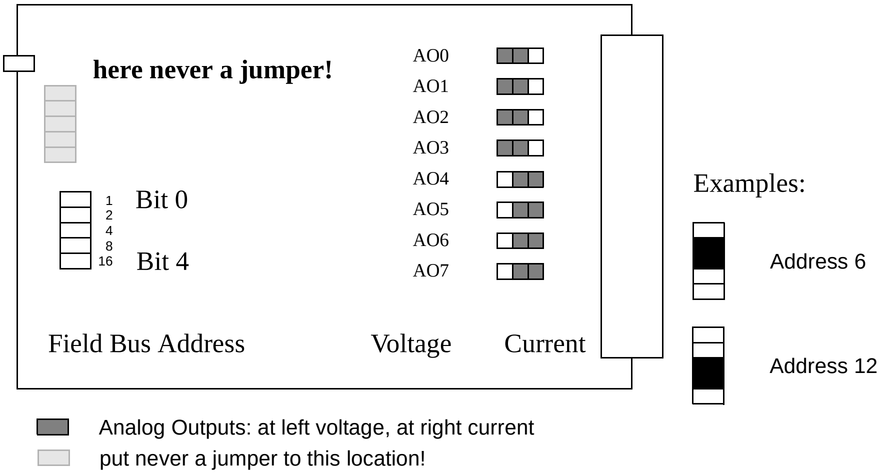


FIGURE 28 S5681: Address- and Output Configuration

1.10.3 Power Supply

The supply voltage (9-30V_{DC}) has to be connected to the DIN connector.

1.10.4 Operation

The LED indicates the operation of the ***S5681***. Fast blinking shows that there is no connection to a master device. Slow blinking shows that ***S5681*** receives data**.**

1.11 Installation and Usage of S9704

S9704 is a powerful controller featuring the VME industrial bus system. ***S9704*** is able to communicate to other ISDN controllers very fast. It is possible to setup ***S9704*** using ISDN as well as the serial interface. It is also possible to send messages to an ISDN Terminal adaptor (***SEAL SLink21***); for these messages or for configuration of ***S9704*** the protocol X.75 is used on the B channel. Beside eight bicolour LEDs which may be adjusted by the user the controller features 4 switches to allow the input of user commands.

1.11.1 Power Supply

The unit is driven by the VME Bus and uses +5V and +/- 12V.

1.11.2 Field Bus

Submodules of the ***S99*** series are connected using the VME Bus.

1.11.3 PC Connection

The 9-Pol. female DSUB connector is used:

Pin	RS232	V.24	Direction	Description
2	BB	104	from S9704	sent data
3	BA	103	to S9704	received data
5	AB	102	both	ground

TABLE 21 S9704: RS232 Connectors

Only pins 2, 3 and 5 are used for the connection to the PC. The baud rate is 9600.

1.11.4 ISDN Connection

The lower 8-pin RJ45 connector is used.

RJ45 - RJ45: Use a 1:1 cable.
RJ45 - TT83: a special cable as shown in the table blow is needed:

Pin	RJ45 (S9704)	RJ45 (PTT)	TT83 (PTT)
TX-	3	3	a1
RX-	4	4	a2
RX+	5	5	b2
TX+	6	6	b1

TABLE 22 S9704: ISDN Connector

1.11.5 Usage

1.11.5.1 LED Display

The eight system LEDs show the internal states of ***SEAL S9704***.

System-state	l	l	User program
ISDN-state	l	l	Klopfen
1. B-channel data-connection	l	l	1. B-channel optional connection
2. B-channel data-connection	l	l	2. B-channel optional connection
User-LED 1	l	l	User-LED 5
User-LED 2	l	l	User-LED 6
User-LED 3	l	l	User-LED 7
User-LED 4	l	l	User-LED 8

FIGURE 29 S9704: LED Display

	red	yellow	green
System	System error	Flash basic FW	Flash extended
ISDN state	no L1	no L2	L1 and L2 OK
1. B channel (Data)		building/closing	connection
2. B channel (Data)		building/closing	connection
User program	none		active
“knocking” (500 ms)	data	optional	configuration
1. B channel (Option)		building/closing	connection
2. B channel (Option)		building/closing	connection

TABLE 23 S9704: LED Display

At the time of distribution the basic firmware is used. The user may load firmware upgrades which is shown by the system LED.

1.11.5.2 Switches

Four switches are located below the LEDs.

1.11.6 Configuration of S9704

To use all features of ***S9704***, several configurations have to be done. In the first place it is necessary to enter the ISDN number.

- **Address:** The address of ***S9704*** is always 1.
- **Cycle rate:** The cycle rate should be 5 ms.
- **MSN:** The multiple subscriber number is used to select a dedicated unit at the ISDN S-Bus.
- **Sub Address Prefix:** Must be the same at all controllers which should communicate with each other and is used as additional security against intruders in case the table of allowed callers is not used.
- **Allowed Caller Table:** Up to 24 allowed callers may be configured.
- **Configuration using ISDN:** It is possible to configure the controller using ISDN only if this option was enabled previously.
- **Event Log:** Up to 1024 events may be logged. Which kind of events are logged may be selected:
  System Events: e.g. start up
  ISDN problem: Layer 1 (physical) or Layer 2 (logical) connection.
  ISDN connection troubles: If a call cannot be done properly.
  ISDN connection: Incoming and outgoing calls.
  User Program: Loading a user program from PC or Flash.

After changing one or more of these parameters, the user software has to be compiled and loaded.

1.12 S5500 Field Bus Card for PC

The ***SEAL S5500*** field bus card is a plug in card for the PC. It is used to connect a controller ***S5701*** or ***S5703*** using the token bus or ***S5610*** using RS485. This allows higher transfer rates than the RS232 interface.

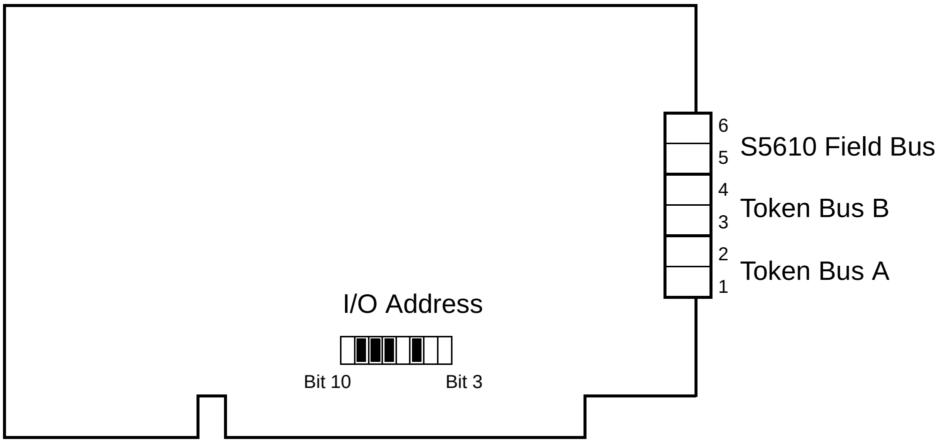


FIGURE 30 Configuration and Connections of the S5500 Field Bus Card

Before putting the card into the PC, a free I/O address in the range from **0x200** to **0x800** has to be configured.

The figure above shows the address **0x3A0** which is the configured address when ***S5500*** is distributed to the user.

In the PC software ***S57901*** *"SEAL Fieldbus Card"* has to be selected instead of one of the COM ports. The software then searches the port addresses in the range from **0x200** to **0x800** and shows all the addresses where an ***S5500*** field bus card may be. Usually this is the address where ***S5500*** really is, otherwise the right address has to be selected. Additionally the field bus address of ***S5500*** may be selected (**2**, **4** or **6**). This is used only if there is more than one (up to three) ***S5500*** at one field bus; then each card must have a unique field bus address. If only one ***S5500*** is used, the value of this address does not matter.

To connect the field bus, the lowest pins (1 and 2) of the 6-pin connector have to be used (*"Token Bus A"*); *"Token Bus B"* is not supported in the current firmware version.

The two upper pins (5 and 6) are used for the RS485 bus in connection with an ***S5610*** configuration.

1.13 General Field Bus Information

Each device which is connected to the token bus must get its own unique token bus address which must be an even number. This address is selected using the appropriate dip switch of the controllers and expansion modules, respectively. The PC field bus card is selected by the software. If there are more than one devices using the same address, these devices will not be able to communicate properly. It may be wise to reset all the devices after such an erroneous configuration was eliminated.

It is also important that no address overlapping occurs. A controller which is adjusted to a token bus address of **8** uses all addresses from **8** to **15** (five internal I/O addresses and two field bus export addresses for analogue and digital values); the next controller or expansion module has to be adjusted to address **16** or higher. A detected address conflict is shown with red letters in the configuration software.

If there are expansion modules of the series ***S59xxE*** connected to a controller, the number of used addresses will be increased appropriate.

It is important that the address of the controller (***S5701*** or ***S5703***) selected by the dip switches is the same as selected using the configuration software. Only then the PC card ***S5500*** will be able to communicate to the controller. If you communicate using the RS232 interface, the selected address does not matter. Additionally it has to be noticed that with wrong address configuration the inputs and outputs of the controller would not work properly.

If the field bus is turned off (all address switches to 0 or all switches to 1) an address value of **1** has to be selected in the configuration software. Naturally ***S5500*** will not be able to talk to the controller using the token bus in this case. If the field bus address is 1 no token bus expansion modules may be entered by the software ***S57901***.

Termination of the field bus twisted pair cable is necessary to ensure proper communications. A resistor of about 150 to 300 Ω is just right.

1.14 Modem Usage

All controllers may be connected to a modem or an ISDN communications device (terminal adaptor). Then they are able to send messages or be called by the user. This features require the wiring shown in the table below:

Signal	Controller Pin	Modem Pin
Security Ground	**1 - GND**	**1 - GND**
Receive Data	**2 - RXD**	**3 - TXD**
Transmit Data	**3 - TXD**	**2 - RXD**
Carrier Detected	**4 - CD**	**8 - CD**
Data Terminal Ready	**5 - DTR**	**4 - RTS** and **20 - DTR**
Signal-Ground	**7 - GND**	**7 - GND**

TABLE 24 Wiring to connect a modem to a controller

The signal *DTR* at pin 5 is active after power on so the controller may be called by the user. The modem has to be adjusted to “Auto Answer” so it takes an incoming call without requiring any commands from the controller.

When the controller wants to transmit a message it puts the dial command to the serial interface. After the modem built the connection the controller detects *CD* at pin 4 and outputs the message.

Afterwards the escape sequence “+++” and “ATH0” will be sent to the modem. Then the signal *DTR* is removed to ensure the disconnection of the telephone line.

It has to be noticed that the time out for a dial command is the same as the time between retries. So if the time between retries is set to 30 seconds, the controller waits for 30 seconds if the first try fails. If an incoming call during this time sets the CD signal, the message will be sent and registered as completed although it may not be put to the right destination. If the PC software is detected within the next 3 seconds the connection will remain until hung up by the PC, otherwise it will be hung up by the controller immediately.

So the time between retries has to be configured to a time out value also if the number of retries is zero.

The **modem** must be configured to ***Echo Off*** and ***Quiet Mode On*** using ATE0Q1, that means no answers to commands. Additionally **Auto Answer** has to be turned on to allow incoming calls. after a power on reset the controller outputs ATS0=1 to allow incoming calls in case the line power was off and the modem has reset.

SEAL SLink21 terminal adaptor and ***SLink41*** PC Card are able to connect controllers and PCs using the **digital network ISDN** without any troubles.

1.15 Device Capabilities

1.15.1 Running User Programs

After configuring a program for the ***S57XX*** with the graphical user interface ***S57901***, you have to download this user program to the ***S57XX***. The operating system of ***S57XX*** starts the program according to the specified run condition.

1.15.2 Data Buffering

You may use up to 32 buffers each containing up to 16 variables. The buffers can be used to collect data over a period of time. There are two kinds of buffers available: a ring buffer which overwrites the oldest values or a buffer which stops collecting data after being filled. The host may initiate a transfer of these buffers at any time and optionally release them to allow further collection of data.

The command *VBS<no>* may be used by a terminal program to dump the content of the specified buffer *<no>*. *<No>* has to be in the range from 1 to 32. *VBC<no>* resets the buffer *<no>*. To select a format for the analog variables, the command *VBF<fmt>* is used: These formats may be selected using the S57901 BUFFER OPTIONS command.

Command	Format	Example	Comment
VBF%g	Standard format	1234.2	smallest format as possible
VBF%19.8f	19 characters at all 4 after decimal point	1234.2000	19 characters are the maximal possible format length
VBF%10.0f	10 characters at all without decimal part	1234	rounded to integer
VBF%8.3e	8 characters exponential format	1.234e+03	in this example the output needs more characters than expected

FIGURE 31 Examples to select a Buffer Format

The first number after '%' is the length of the variable, the (optional) second number is the number of characters after the decimal point.

1.15.3 Output Buffers

These are used to set variables which are not connected to an output. To load these buffers a file like one generated by the ***S57901*** software is required.

1.15.4 Time Dependent Running of Program Segments

For each program segment you may specify a run condition using the well known cron format. A description of this format is given later in this manual.

It is possible to run a whole segment slower than the basic cycle time.

CHAPTER 2 Function Block Description

2.1 Input/Output Functions

The PC software S57PROG handles input and output function blocks in the same way as all the other function blocks. The I/O function blocks are named FncIO<no>. Currently the following I/O functions are available:

DIGITAL INPUT galvanically isolated
DIGITAL INPUT TTL

DIGITAL OUTPUT relay
DIGITAL OUTPUT driver

ANALOG INPUT differential +/-10V
ANALOG INPUT differential +/-20mA
ANALOG INPUT user configure-able

COUNTER with quadrature decoder

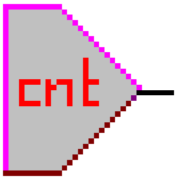
ANALOG OUTPUT +/-10V 16bit
ANALOG OUTPUT +/-20mA 16bit
ANALOG OUTPUT frequency
ANALOG OUTPUT user configure-able

Each I/O function block has two parameters which should be configured by the user:

- the cycle time to read respectively write the I/O function
- the channel of the I/O device

After such a function block is inserted, the cycle time is set to the fastest cycle time available for this device; this is usually the basic cycle time of that device. The channel number is set to the next free channel. If another compatible device or channel has to be used, the user must change the channel parameter appropriate.

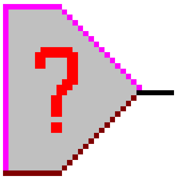
2.1.1 Counter Input for Quadratur-Decoder



output:	one analog output
purpose:	reading of pulse counter or velocity.

For each quadratur decoder 2 such inputs may be used. Input 0 counts the pulses of decoder 1, input 1 shows the number of pulses per second. Input 2 and 3 are used in the same way for decoder 2. You should use long time masks for the velocity inputs if you count slow pulse rates to get continuous input signals. The connection of the encoders is shown in the technical description of ***S5701*** and ***S5703***.

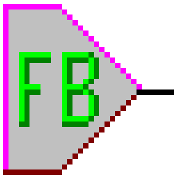
2.1.2 Analog Input, User Specified



output:	one analog output

These inputs has to be adjusted appropriate. The jumpers must be set to the desired places and the software must be adjusted for the right type of input.

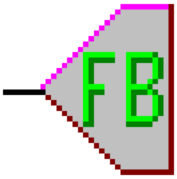
2.1.3 Fieldbus Input



output:	one digital or one analog output
purpose:	reading of a value from another controller at the field bus.

From each other controller connected to the field bus up to 16 digital and up to 16 analog variables may be received. You must adjust the address of the other controller and the channel number. For analog values the field bus address is used, for digital values the field bus address + 1 is used.

2.1.4 Fieldbus Output

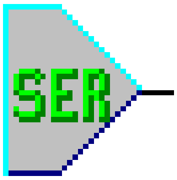


input:	one digital or one analog input
purpose:	sending a value to another controller at the field bus.

Each controller ***S5701*** or ***S5703*** can export up to 16 digital and up to 16 analog values to the field bus. which may received by other controllers

.

2.1.5 Serial Input



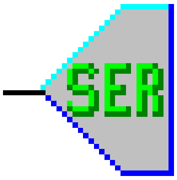
library entry:	FncRSDI, FncRSAI
length:	16 bytes
outputs:	one digital respectively one analog output
purpose:	Setting of a digital or an analog variable.

These functions get an identification number automatically after being inserted into the user program. This number is used to identify the corresponding variable.

Example of a command being entered at the serial port:

VSD <no> 0	Clearing of the digital output <no>
VSD <no> 1	Setting of the digital output <no>
VSA <no> 3.14	Setting of the analog output <no>

2.1.6 Serial Output



library entry:	FncRSDO, FncRSAO
length:	16 bytes
inputs:	one digital respectively one analog input
purpose:	Request of a value using a serial port.

These functions get an identification number automatically after being inserted into the user program.

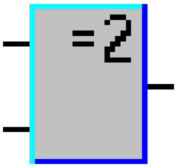
Example:

VRD <no>	Request of a digital value
VD <no> 1	Answer of the device
VRA <no>	Request of an analog value
VA <no> 8.4	Answer of the device

2.2 Logical Functions

Generally digital values 0 are FALSE and 1 are TRUE.

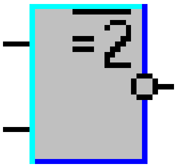
2.2.1 Logical AND



library entry:	FncAnd
length:	12 bytes
inputs:	two digital inputs
outputs:	one digital output
purpose:	logical AND of the two inputs

input 1	input 2	output
0	0	0
1	0	0
0	1	0
1	1	1

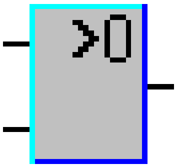
2.2.2 Logical NAND



library entry:	FncNand
length:	12 bytes
inputs:	two digital inputs
outputs:	one digital output
purpose:	logical NAND of the two inputs

input 1	input 2	output
0	0	1
0	1	1
1	0	1
1	1	0

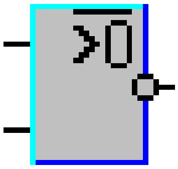
2.2.3 Logical OR



library entry:	FncOr
length:	12 bytes
inputs:	two digital inputs
outputs:	one digital output
purpose:	logical OR of the two inputs

input 1	input 2	output
0	0	0
0	1	1
1	0	1
1	1	1

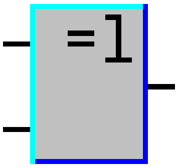
2.2.4 Logical NOR



library entry: FncNor
length: 12 bytes
inputs: two digital inputs
outputs: one digital output
purpose: logical NOR of the two inputs

input 1	input 2	output
0	0	1
0	1	0
1	0	0
1	1	0

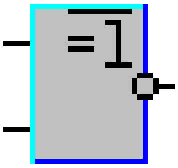
2.2.5 Logical EXOR



library entry: FncExor
length: 12 bytes
inputs: two digital inputs
outputs: one digital output
purpose: logical EXOR of the two inputs

input 1	input 2	output
0	0	0
0	1	1
1	0	1
1	1	0

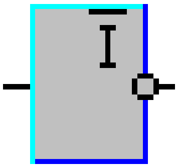
2.2.6 Logical EXNOR



library entry: FncExnor
length: 12 bytes
inputs: two digital inputs
outputs: one digital output
purpose: logical EXNOR of the two inputs

input 1	input 2	output
0	0	1
0	1	0
1	0	0
1	1	1

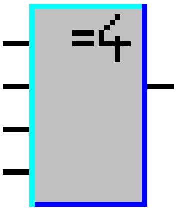
2.2.7 Logical NOT



library entry: FncNot
length: 10 bytes
inputs: one digital input
outputs: one digital output
purpose: logical NOT of the input

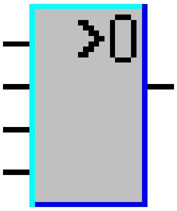
input	output
0	1
1	0

2.2.8 Logical AND with 4 or 8 Inputs



library entry: Fnc4And. Fnc8And
length: 16 or 24 bytes, respectively
inputs: 4 or 8 digital inputs, respectively
outputs: one digital output
purpose: logical AND of the inputs

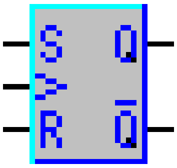
2.2.9 Logical OR with 4 or 8 Inputs



library entry: Fnc4Or. Fnc8Or
length: 16 or 24 bytes, respectively
inputs: 4 or 8 digital inputs, respectively
outputs: one digital output
purpose: logical OR of the inputs

2.3 Flip Flops

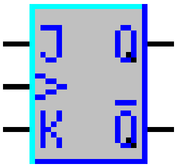
2.3.1 SR Flip Flop



library entry:	FncSRFlip
length:	18 bytes
inputs:	3 digital inputs
outputs:	2 digital output
purpose:	behave like an SR FLip Flop

At the rising edge of the clock input the inputs *s* and *r* are evaluated. Then the outputs *q* and *inv* are set accordingly.

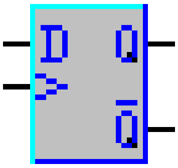
2.3.2 JK Flip Flop



library entry:	FncJKFlip
length:	18 bytes
inputs:	3 digital inputs
outputs:	2 digital output
purpose:	behave like a JK FLip Flop

At the rising edge of the clock input the inputs *j* and *k* are evaluated. At the next cycle the outputs *q* and *inv* are set accordingly.

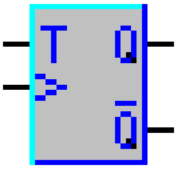
2.3.3 D Flip Flop



library entry:	FncDFlip
length:	16 bytes
inputs:	2 digital inputs
outputs:	2 digital output
purpose:	behave like a D FLip Flop

At the rising edge of the clock input the input *d* is evaluated. At the next cycle the outputs are set accordingly.

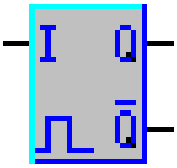
2.3.4 T Flip Flop



library entry:	FncTFlip
length:	16 bytes
inputs:	2 digital inputs
outputs:	2 digital output
purpose:	behave like a T FLip Flop

At the rising edge of the clock input the input *t* is evaluated. If it is true, the current value of the output is saved and at the next cycle the outputs are set accordingly.

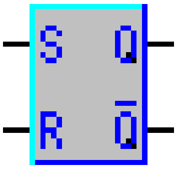
2.3.5 Monostable Flip Flop



library entry:	FncMonoFlip
length:	22 bytes
inputs:	1 digital input
outputs:	2 digital outputs
parameters:	the number of cycles to hold the output
purpose:	behave like a Mono Flop

After the input changes to TRUE, the output also becomes TRUE. After the input changed to FALSE, the output keeps TRUE for the selected number of cycles.

2.3.6 SR Flip Flop without Clock

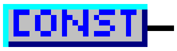


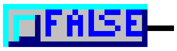
library entry:	FncSRN
length:	14 bytes
inputs:	2 digital inputs
outputs:	2 digital output
purpose:	behave like an SR FLip Flop

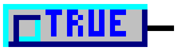
If the input s is set, the output q is set and $\bar{q}$ is cleared. If the input r is set, the output q is cleared and $\bar{q}$ is set. The input r has the higher priority.

2.4 Further Digital Functions

2.4.1 Digital Constant

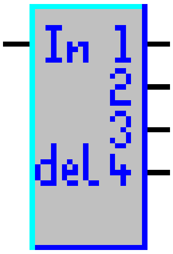






library entry:	FncDgConst
length:	10 bytes
outputs:	1 digital output
parameters:	TRUE or FALSE
purpose:	set the output to a constant TRUE or FALSE

2.4.2 Digital Delay

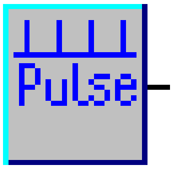


library entry:	FncDgDelay
length:	40 bytes
inputs:	1 digital inputs
outputs:	4 digital outputs
parameter:	no. of cycles to delay
purpose:	delay the digital input

On output 1 the input signal is delayed the specified number of cycles. The other outputs delay 2, 4 and 8 times the number of specified cycles, respectively. Note that the input is scanned only at the specified intervals so fast changes of the input signal do not appear on the outputs.

output	delay
output 1	1 ∗ no of specified cycles
output 2	2 ∗ no of specified cycles
output 3	4 ∗ no of specified cycles
output 4	8 ∗ no of specified cycles

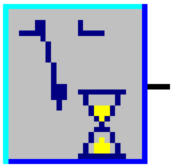
2.4.3 Pulse Generator



library entry:	FncPlsGen
length:	16 bytes
outputs:	1 digital output
parameter:	no. of cycles between pulses
purpose:	generate pulses

This function sets the output TRUE for the duration of 1 cycle. The time between these pulses is defined by the parameter.

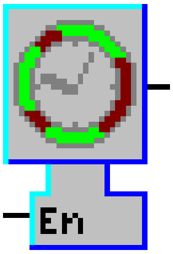
2.4.4 System Time Switch



library entry:	FncTmrSw
length:	24 bytes
outputs:	1 digital output
parameter:	system time to switch on and off
purpose:	generate TRUE between two times

This function sets the output TRUE at the specified system time “on”, the output is set back to FALSE at the specified system time “off”.

2.4.5 Timer

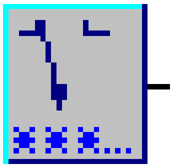


library entry:	FncTimer
length:	138 bytes
input:	1 digital input
output:	1 digital output
parameter:	times to turn on the output.
purpose:	programmable timer switch

This function turns on the output at the desired times. At all other times the output is turned off. If the enable input is connected and off, the output will neither be set nor cleared to allow parallel connection of several timers (tristate output).

range	accuracy
1 hour	1 minute
1 day	5 minutes
1 week	1 hour
1 month	1 hour
1 year	1 day

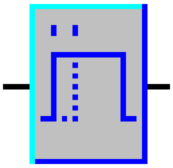
2.4.6 Cron Switch



library entry:	FncCronSw
length:	72 bytes
outputs:	1 digital output
parameter:	cron time to switch on and off
purpose:	generate TRUE at specified times

This function sets the output TRUE if the time matches the specified cron time. For a description of the cron format see “Time Dependent Starting of Program Segments” on page 3-11.

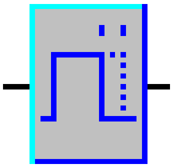
2.4.7 Digital Delay for Switching ON



library entry:	FncOnDel
length:	18 bytes
inputs:	1 digital inputs
outputs:	1 digital outputs
parameter:	no. of cycles to delay
purpose:	delay the rising edge of the digital input

When the input changes from LOW to HIGH, the output change is delayed by the specified number of cycles. When the input changes from HIGH to LOW, the output follows immediately.

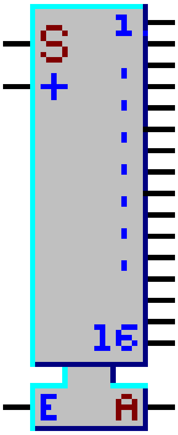
2.4.8 Digital Delay for Switching OFF



library entry:	FncOffDel
length:	18 bytes
inputs:	1 digital inputs
outputs:	1 digital outputs
parameter:	no. of cycles to delay
purpose:	delay the falling edge of the digital input

When the input changes from HIGH to LOW, the output change is delayed by the specified number of cycles. When the input changes from LOW to HIGH, the output follows immediately.

2.4.9 Flow Control

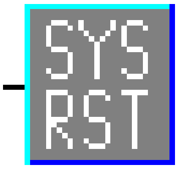


library entry:	FncCtl
length:	52 bytes
inputs:	1 analog input 2 digital inputs
outputs:	1 analog output 16 digital outputs
purpose:	flow control of digital outputs

If the enable input “*E*” is off, all outputs are zero. At the rising edge of the enable input the value of the set input “*S*” is read, if connected. Then the corresponding output is set, otherwise the first output is set. At each rising edge of the input “+” the current output is cleared and the next output is set until the last output is reached or an output is not connected. Then the function block continues with the first output. The analog output “*A*” contains the number of the active output or zero if no output is set.

2.5 System Near Functions

2.5.1 System Reset

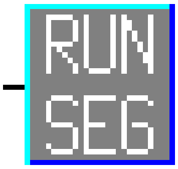


library entry:	FncSysRS
length:	20 bytes
inputs:	1 digital input
parameter:	the parts of the system to be reset
purpose:	system initialization

At the rising edge of the input the desired reset of the system is performed:

Reset of the buffers
all buffers
only write buffers
only read buffers
a specified buffer
Reset of the function blocks
Reset of the variables
Reset of the counter inputs
Activation of the power on function blocks

2.5.2 Start of Another Segment

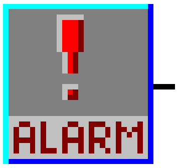


library entry:	FncSegm
length:	16 bytes
inputs:	1 digital input
parameter:	which segment should be started how
purpose:	start or stop other program segments

If the input variable becomes TRUE the segment will be started or stopped:

start segment, ignore cron condition:
the segment runs if the input is TRUE.
stop segment ignore cron condition:
the segment runs if the input is FALSE.
run only if cron condition fulfilled:
segment runs if input is TRUE
***and** the cron condition is fulfilled.*
run also if cron condition fulfilled:
segment runs if input is TRUE
***or** the cron condition is fulfilled.*

2.5.3 Alarm



library entry:	FncWarn
length:	18 bytes
outputs:	1 digital output
parameter:	Alarm condition and supplement value
purpose:	To inform about several system states

The output becomes TRUE at the occurrence of the selected condition:

Cycle time becomes greater than a given limit
A buffer is full
Communication to a specified field bus device failed
A modem message could not be sent

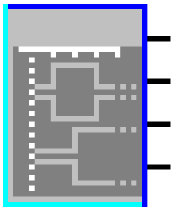
If the device features an ISDN interface, also one of the following may be selected:

No physical or logical ISDN link
ISDN connection troubles
outgoing ISDN connections
incoming ISDN connections

Example:

To allow endless buffering of data, this function may be used to switch between two buffers. If the first buffer becomes full, the data is buffered in the second buffer while the content of the first one is transmitted to the PC.

2.5.4 Power On Pulses



library entry:	FncPwrOn
length:	20 bytes
outputs:	4 digital outputs
parameter:	number of cycles to generate the power on pulses
purpose:	generate signals after power on

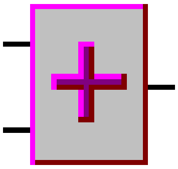
This function generates the signals indicated by the icon after power on as well as after each loading of the program. These signals may be used to reset the user software. After the number of cycles specified by the parameter, the first two outputs will change their value. The same time later the second two outputs change their value. After another period of this time the first two outputs are reset to their initial value.

Cycles after Power On	Output 1	Output 2	Output 3	Output 4
< (n+1)	0	1	0	1
(n + 1) .. (2*n + 2)	0	1	1	0
(2*n + 3) .. (3*n + 3)	1	0	1	0
> (3*n + 3)	1	0	0	1

The table above shows the states of the outputs with n as parameter “number of cycles”.

2.6 Mathematical Functions

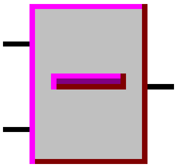
2.6.1 Addition



library entry:	FncAdd
length:	12 bytes
inputs:	2 analog inputs
outputs:	1 analog output
purpose:	add two analog values

$$output = input1 + input2$$

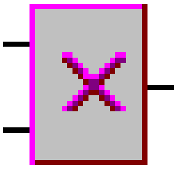
2.6.2 Subtraction



library entry:	FncSub
length:	12 bytes
inputs:	2 analog inputs
outputs:	1 analog output
purpose:	subtract two analog values

$$output = input1 - input2$$

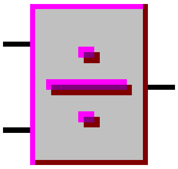
2.6.3 Multiplication



library entry:	FncAdd
length:	12 bytes
inputs:	2 analog inputs
outputs:	1 analog output
purpose:	multiply two analog values

$$output = input1 \times input2$$

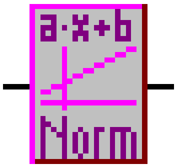
2.6.4 Division



library entry:	FncDiv
length:	12 bytes
inputs:	2 analog inputs
outputs:	1 analog output
purpose:	divide two analog values

$$output = (input1)/(input2)$$

2.6.5 Linear Transformation

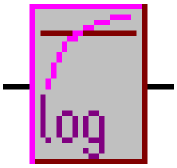


library entry:	FncNormAB
length:	22 bytes
inputs:	1 analog input
outputs:	1 analog output
parameters:	time mask, A and B
purpose:	transform a measurement value

This function performs a linear transformation of the input at the specified cycles.

$$output = A \times input + B$$

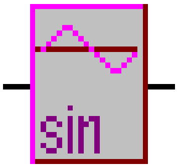
2.6.6 Logarithm



library entry:	FncLog
length:	10 bytes
inputs:	1 analog input
outputs:	1 analog output
purpose:	calculate the logarithm of an analog value

$$output = \log(input)$$

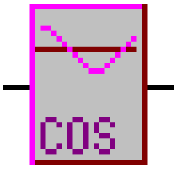
2.6.7 Sine



library entry:	FncSin
length:	10 bytes
inputs:	1 analog input
outputs:	1 analog output
purpose:	calculate the sine of an analog value

$$output = \sin(input)$$

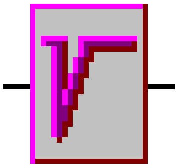
2.6.8 Cosine



library entry:	FncCos
length:	10 bytes
inputs:	1 analog input
outputs:	1 analog output
purpose:	calculate the cosine of an analog value

$$output = \cos(input)$$

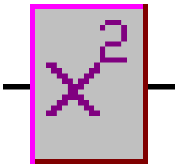
2.6.9 Square Root



library entry:	FncSqrt
length:	10 bytes
inputs:	1 analog input
outputs:	1 analog output
purpose:	calculate the square root of an analog value

$$\mathit{output} = \sqrt{\mathit{input}}$$

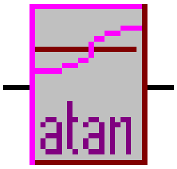
2.6.10 Square



library entry:	FncX2
length:	10 bytes
inputs:	1 analog input
outputs:	1 analog output
purpose:	calculate the square of an analog value

$$\mathit{output} = (\mathit{input})^2$$

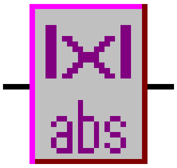
2.6.11 Arcus Tangens



library entry:	FncAtan
length:	10 bytes
inputs:	1 analog input
outputs:	1 analog output
purpose:	calculate the arcus tangens of an analog value

$$\mathit{output} = \operatorname{atan}(\mathit{input})$$

2.6.12 Absolute Value



library entry:	FncAbs
length:	10 bytes
inputs:	1 analog input
outputs:	1 analog output
purpose:	calculate the absolute value of an analog value

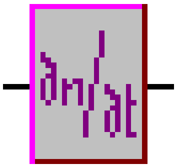
$$\mathit{output} = |\mathit{input}|$$

2.6.13 Analog Constant

library entry:	FncAnConst
length:	10 bytes
outputs:	1 analog output
parameters:	the analog value to be set
purpose:	set the output to a constant value

$$output = parameter$$

2.6.14 Differenciator

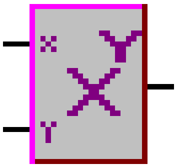


library entry:	FncDNNDT
length:	14 bytes
inputs:	1 analog input
outputs:	1 analog output
purpose:	calculate the changing of an analog value

$$output = (\partial n)/(\partial t)$$

The output is the changing of the input per second. It is calculated each cycle.

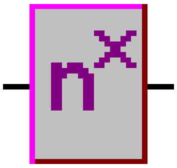
2.6.15 Exponential Function with Base as Variable



library entry:	FncPow
length:	12 bytes
inputs:	2 analog inputs
outputs:	1 analog output
purpose:	calculation of x to the power of y

$$output = x^y$$

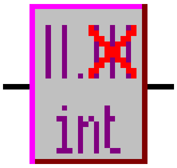
2.6.16 Exponential Function with Constant Base



library entry:	FncExpo
length:	14 bytes
inputs:	1 analog input
outputs:	1 analog output
parameter:	base
purpose:	calculation of n to the power of x

$$output = n^x$$

2.6.17 Integer Value of an Analog Variable



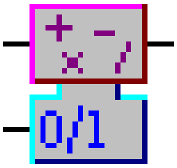
library entry: FncInt
length: 10 bytes
inputs: 1 analog input
outputs: 1 analog output
purpose: calculation of the integer part of an analog variable

$$output = floor(input)$$

The value of the analog input is truncated to the next smaller integer value, e.g.:

Input Value	Output Value
0	0
3.14	3
-5.23	-5

2.6.18 Calculation in Dependence of Digital Input

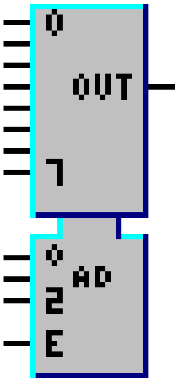


library entry: Fnccreq
length: 24 bytes
inputs: 1 analog input
1 digital input
outputs: 1 analog output
parameters: kind of operation, operators
purpose: calculation dependent of digital value

For each state of the digital input the operation (addition, multiplication) and the value of the operator may be selected. So it is possible to e.g. invert the signal at one time and add an offset at another time.

2.7 Multiplexer Functions

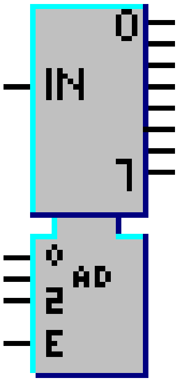
2.7.1 Digital Multiplexer



library entry:	FncDgMux
length:	32 bytes
inputs:	12 digital inputs
outputs:	one digital output
purpose:	select one of eight inputs

The binary value of the 3 address inputs refers to one of the eight digital inputs. If the enable input is TRUE, the output is set due to the value of the selected input. Otherwise the output is left unchanged. This allows to connect more than one multiplexer output together (tristate output).

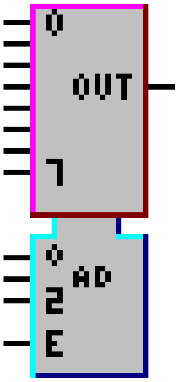
2.7.2 Digital Demultiplexer



library entry:	FncDgDemux
length:	32 bytes
inputs:	5 digital inputs
outputs:	8 digital output
purpose:	select one of eight outputs

The binary value of the 3 address inputs refers to one of the eight digital outputs. If the enable input is TRUE, the selected output is set due to the value of the input. Otherwise the output is left unchanged. This allows to connect more than one demultiplexer output together (tristate output).

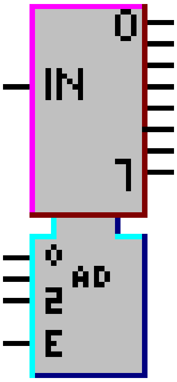
2.7.3 Analog Multiplexer



library entry:	FncAnMux
length:	32 bytes
inputs:	8 analog inputs
	4 digital inputs
outputs:	one analog output
purpose:	select one of eight inputs

The binary value of the 3 address inputs refers to one of the eight digital inputs. If the enable input is TRUE, the output is set due to the value of the selected input. Otherwise the output is left unchanged. This allows to connect more than one multiplexer output together (tristate output).

2.7.4 Analog Demultiplexer



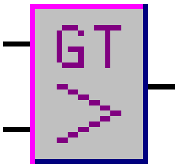
library entry:	FncAnDemux
length:	32 bytes
inputs:	1 analog input 4 digital inputs
outputs:	8 analog output (tristate)
purpose:	select one of eight outputs

The binary value of the 3 address inputs refers to one of the eight digital outputs. If the enable input is TRUE, the selected output is set due to the value of the input. Otherwise the output is left unchanged. This allows to connect more than one demultiplexer output together (tristate output).

2.8 Compare Functions

All compare functions have their input 1 above their input 2.

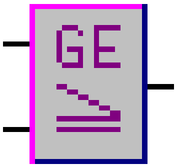
2.8.1 Greater Than



library entry:	FncGT
length:	12 bytes
inputs:	2 analog inputs
outputs:	1 digital output
purpose:	compare two analog values

inputs	output
in1 > in2	1
in1 ≤ in2	0

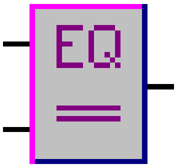
2.8.2 Greater or Equal



library entry:	FncGE
length:	12 bytes
inputs:	2 analog inputs
outputs:	1 digital output
purpose:	compare two analog values

inputs	output
in1 Š in2	1
in1 < in2	0

2.8.3 Equal

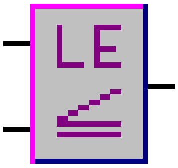


library entry:	FncEQ
length:	12 bytes
inputs:	2 analog inputs
outputs:	1 digital output
purpose:	compare two analog values

inputs	output
in1 ≡ in2	1
in1 ≠ in2	0

This function should be used with care because two floating values seldom appear to be equal. Usually the functions FncGT and FncLT should be used to build a comparing function.

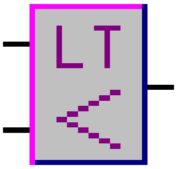
2.8.4 Less or Equal



library entry:	FncLE
length:	12 bytes
inputs:	2 analog inputs
outputs:	1 digital output
purpose:	compare two analog values

inputs	output
in1 ≤ in2	1
in1 > in2	0

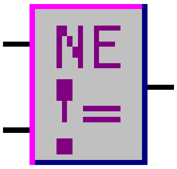
2.8.5 Less Than



library entry:	FncLT
length:	12 bytes
inputs:	2 analog inputs
outputs:	1 digital output
purpose:	compare two analog values

inputs	output
in1 < in2	1
in1 Š in2	0

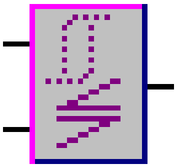
2.8.6 Not Equal



library entry:	FncNE
length:	12 bytes
inputs:	2 analog inputs
outputs:	1 digital output
purpose:	compare two analog values

inputs	output
in1 ≠ in2	1
in1 ≡ in2	0

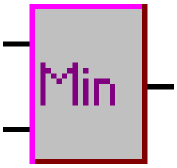
2.8.7 Compare with Hysteresis



library entry:	Fnccohy
length:	18 bytes
inputs:	2 analog inputs
outputs:	1 digital output
parameters:	hysteresis, operation
purpose:	compare two values

The output depends of the input values and of the kind of operation. The hysteresis avoids fast output changes if the input values are nearly the same.

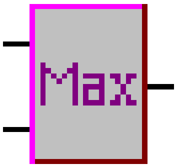
2.8.8 Minimum



library entry:	FncMin
length:	12 bytes
inputs:	2 analog inputs
outputs:	1 analog output
purpose:	select minimum of two inputs

inputs	output
in1 < in2	input 1
in1 > in2	input 2

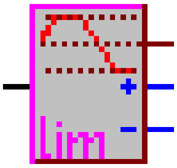
2.8.9 Maximum



library entry:	FncMax
length:	12 bytes
inputs:	2 analog inputs
outputs:	1 analog output
purpose:	select maximum of two inputs

inputs	output
in1 > in2	input 1
in1 < in2	input 2

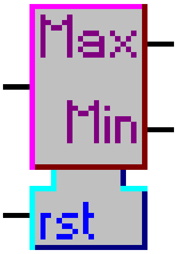
2.8.10 Limiter



library entry:	FncLim
length:	22 bytes
inputs:	1 analog input
outputs:	1 analog output 2 digital output
parameters:	Maximum Minimum
purpose:	limiting of the input value

If the input exceeds the maximum parameter, the digital output + is set and the output value becomes the maximum parameter. If the input value is less than the minimum parameter, the digital output - is set and the output value equals the minimum parameter.

2.8.11 Storage of Extreme Values

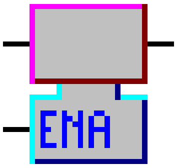


library entry:	FncStore
length:	14 bytes
inputs:	1 analog input 1 digital input (reset)
outputs:	2 analog outputs
purpose:	Storage of the maximum and of the minimum

The analog output *Max* stores the maximal value of the input signal while the analog output *Min* stores the minimal value. If the rst input is TRUE, both *Min* and *Max* output follow the input signal.

2.9 Switches

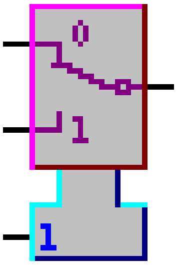
2.9.1 Enable



library entry:	FncEnable
length:	12 bytes
inputs:	1 analog inputs 1 digital input
outputs:	1 analog output (tristate)
purpose:	enable an analog output

input ENA	output
TRUE	≡ input
FALSE	tristate

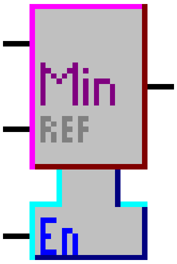
2.9.2 Analog Switch



library entry:	FncAnaSw
length:	14 bytes
inputs:	2 analog inputs 1 digital input
outputs:	1 analog output
purpose:	select an analog output

input ENA	output
TRUE	input 1
FALSE	input 0

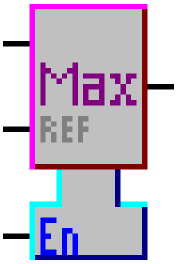
2.9.3 Minimum with Enable



library entry:	FncMinEna
length:	14 bytes
inputs:	2 analog inputs 1 digital input
outputs:	1 analog output (tristate)
purpose:	select minimum

enable	analog inputs	output
TRUE	in1 < in_{ref}	in1
TRUE	in1 > in_{ref}	in_{ref}
FALSE	*don't care*	tristate

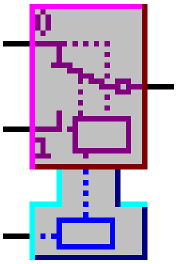
2.9.4 Maximum with Enable



library entry:	FncMaxEna
length:	14 bytes
inputs:	2 analog inputs
	1 digital input
outputs:	1 analog output (tristate)
purpose:	select maximum

enable	analog inputs	output
TRUE	$in1 > in_{ref}$	in1
TRUE	$in1 < in_{ref}$	in_{ref}
FALSE	*don't care*	tristate

2.9.5 Soft Switch



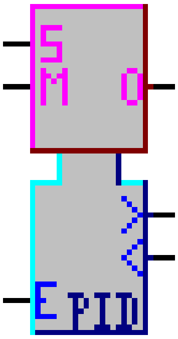
library entry:	FncSoftSw
length:	96 bytes
inputs:	2 analog inputs
	1 digital input
outputs:	1 analog output
parameters:	delay time
	number of filter points
	maximum difference for filtering
	time mask for filter
purpose:	select an analog output

input ENA	output
TRUE	input 1
FALSE	input 0

This function implements a filter if it switches the inputs. As long as the difference between the output value and the value of the selected input exceeds the parameter "diff", the input is filtered before it is put to the output. This filter may be adjusted to use 2, 4, 8 or 16 steps until the output becomes equal to the input. It behaves like the function block "Filter" on page 2-35 which always uses eight steps. After the specified number of cycles after a switching, specified by the parameter "delay", the output follows the selected input.

2.10 Regulators

2.10.1 PID Regulator



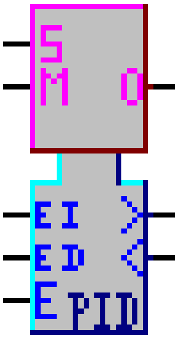
library entry:	FncPID
length:	50 bytes
inputs:	2 analog inputs
	1 digital input (enable)
outputs:	1 analog output (tristate)
	2 digital outputs (tristate)
purpose:	This function behaves like a PID regulator.

“M” is the measured analog input, and “s” is the analog input representing the value that should be achieved. The digital input “e” is used to enable the PID regulator. If the output value is higher than the defined upper limit (constant **ul**), the digital output “u” is set. If the output value is lower than the defined lower limit (constant **ll**), the digital output “l” is set. The output “o” is calculated according to the following formula:

$o = proportional + integral + der$
$proportional = (s - m) \times p$
$integral = integral + (s - m) \times i$
$der = ((s - m) - dp) \times d$
$dp = s - m$

with:
p - proportional value specified as a constant
i - integral value specified as a constant
d - derivative value specified as a constant

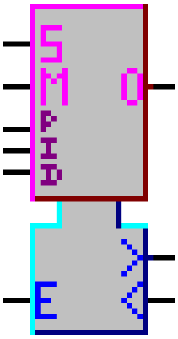
2.10.2 PID Regulator with Enable I and D



library entry:	FncPID_1D
length:	54 bytes
inputs:	2 analog inputs
	2 digital inputs (enable)
outputs:	1 analog output (tristate)
	2 digital outputs (tristate)
purpose:	This function behaves like a PID regulator.

The functionality of this regulator is the same as the one described previously. This function allows to disable the integral or the derivational part of the regulator by putting FALSE to the respective enable input.

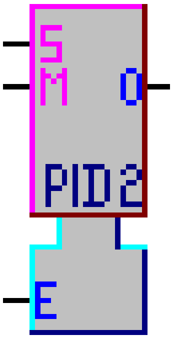
2.10.3 Parametric PID Regulator



library entry:	FncPID_2
length:	54 bytes
inputs:	5 analog inputs 1 digital input (enable)
outputs:	1 analog output (tristate) 2 digital outputs (tristate)
purpose:	This function behaves like a PID regulator.

The functionality of this regulator is the same as the one described previously. This function takes the parameters for proportional, integral and derivational part from the respective analog inputs. This allows to build adaptive regulators.

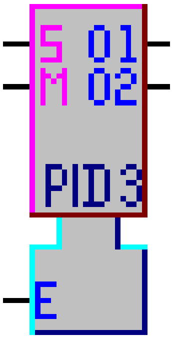
2.10.4 2 Point PID Regulator



library entry:	Fnc2PID
length:	46 bytes
inputs:	2 analog inputs 1 digital input (enable)
outputs:	1 digital output (tristate)
purpose:	This function behaves like a PID regulator.

This function produces a digital output o with PID behaviour. “M” is a measured analog input, and “s” is the analog input representing the value that should be achieved. The digital input “e” is used to enable the PID regulator.
This function block is not implemented in the current firmware version.

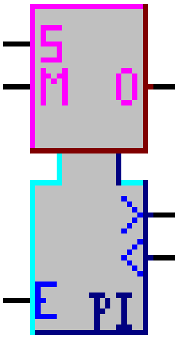
2.10.5 3 Point PID Regulator



library entry:	Fnc3PID
length:	74 bytes
inputs:	2 analog inputs 1 digital input (enable)
outputs:	2 digital outputs (tristate)
purpose:	This function behaves like a PID regulator.

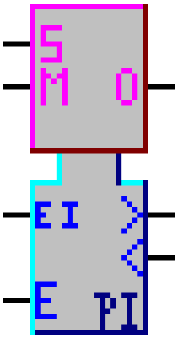
This function produces two digital outputs with 3 point PID behaviour. “M” is a measured analog input, and “s” is the analog input representing the value that should be achieved. The digital input “e” is used to enable the PID regulator.

2.10.6 PI Regulator



library entry:	FncPI
length:	42 bytes
inputs:	2 analog inputs 1 digital input (enable)
outputs:	1 analog output (tristate) 2 digital outputs (tristate)
purpose:	Behaves like a PID regulator with no derivational part.

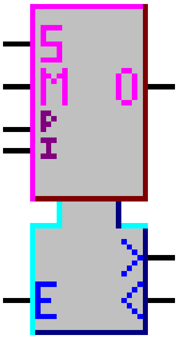
2.10.7 PI Regulator with Enable I



library entry:	FncPI_1
length:	44 bytes
inputs:	2 analog inputs 2 digital inputs (enable)
outputs:	1 analog output (tristate) 2 digital outputs (tristate)
purpose:	Behaves like a PID regulator with no derivational part.

The integral part of this regulator may be disabled by a logical FALSE (0) at the appropriate enable input.

2.10.8 Parametric PI Regulator

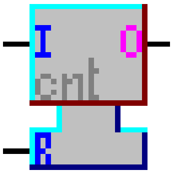


library entry:	FncPI_2
length:	38 bytes
inputs:	4 analog inputs 1 digital input (enable)
outputs:	1 analog output (tristate) 2 digital outputs (tristate)
purpose:	Behaves like a PID regulator with no derivational part.

The functionality of this regulator is the same as the one described previously. This function takes the parameters for proportional and integral part from the respective analog inputs. This allows to build adaptive regulators.

2.11 General Functions

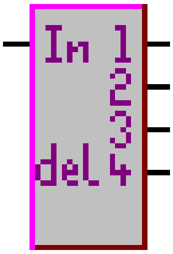
2.11.1 Counter



library entry: FncCounter
length: 14 bytes
inputs: 2 digital inputs
outputs: 1 analog output
purpose:

This function counts the clock cycles at which the input “I” is TRUE. If the reset input “R” is TRUE, the output is reset to 0.

2.11.2 Analog Delay

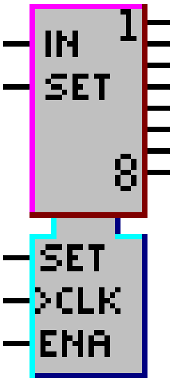


library entry: FncAnDelay
length: 40 bytes
inputs: 1 analog inputs
outputs: 4 analog outputs
parameter: no. of cycles to delay
purpose: delay the analog input

On output 1 the input signal is delayed the specified number of cycles. The other outputs delay 2, 4 and 8 times the number of specified cycles, respectively. Note that the input is scanned only at the specified intervals so fast changes of the input signal do not appear on the outputs.

output	delay
output 1	1 ∗ no of specified cycles
output 2	2 ∗ no of specified cycles
output 3	4 ∗ no of specified cycles
output 4	8 ∗ no of specified cycles

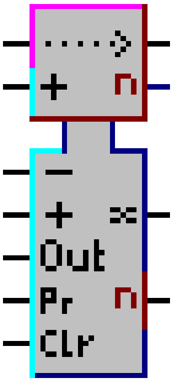
2.11.3 Analog Shift Register



library entry: FncShift
length: 38 bytes
inputs: 2 analog inputs
3 digital inputs
outputs: 8 analog outputs
parameter: reset at state or rising edge of the digital set input
purpose: delay of the analog input signal

At each rising edge of the digital input *clk* the input signal will be shifted through the function block. That means that *output 8* gets the value of *output 7*, *output 7* gets the value of *output 6* and so on. Finally *output 1* gets the value of the analog input *in*. If *ena* is not TRUE, the outputs are tristate. If the digital input *set* is TRUE or gets a rising edge, all outputs are set to the value of the analog input *set*. The *set* input has higher priority than the *clock* input.

2.11.4 Analog Dual-Port Shift Register / Stack

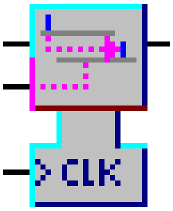


library entry:	FncASR
length:	174 bytes
inputs:	1 analog input 1 digital input to store data 5 digital inputs to read data
outputs:	3 analog outputs 1 digital output
purpose:	store up to 16 values

The input *clr* clears the memory.Each rising edge of the upper input “+” stores the input value and increments the upper output “*n*” until the maximum of 16 is reached.

At the rising edge of the input *out* the last stored value is put to the output. Each rising edge of the input “-” outputs the previous value and increments the lower output “*n*” until the last value is reached. Then the output “*x*” is also set. The inputs “+” (lower), and *pr* are not implemented in the moment.

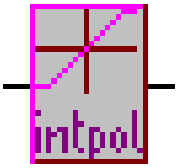
2.11.5 Digital Shift Register



library entry:	FncDgShi
length:	146 bytes
inputs:	1 analog input 2 digital inputs
outputs:	1 digital output
purpose:	Delay of the digital input due to the value of the analog input.

At each rising edge of the digital input *clk* the digital input signal is shifted through the function block. The analog input defines after how many clock cycles the input is put to the output. If the analog input is less or equal to 0, the input is put to the output at the next rising edge of the clock input. If the analog input is greater than 0, the input is delayed for 1 to a maximum of 1024 clock cycles.

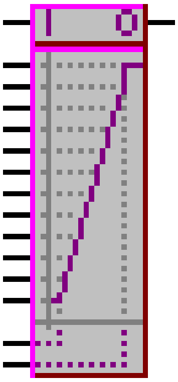
2.11.6 Interpolation



library entry:	FncInterp
length:	90 bytes
inputs:	1 analog input
outputs:	1 analog output
parameters:	10 points for interpolation curve
purpose:	interpolation of input

10 x values and 10 y values specify 10 points. The input value is linear interpolated between the two neighboured x values.

2.11.7 Parametric Interpolation Type 1

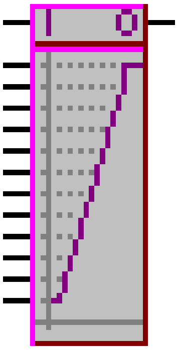


library entry:	FncInter1
length:	42 bytes
inputs:	15 analog input
outputs:	1 analog output
parameters:	time mask
purpose:	interpolation of input

This type of interpolation function calculates 12 equally distant values for the x axis between the values from the x inputs. For each calculated x value the y value is taken from the appropriate input. The output is calculated at the specified times.

Note that if the difference between the two x inputs is zero, the output becomes undefined.

2.11.8 Parametric Interpolation Type 2

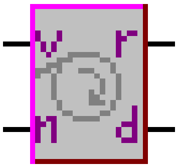


library entry:	FncInter2
length:	100 bytes
inputs:	13 analog input
outputs:	1 analog output
parameters:	time mask and X_{MIN}, X_{MAX}
purpose:	interpolation of input

This type of interpolation function calculates 12 equally distant values for the x axis from the parameters X_{MIN} and X_{MAX} the first time this function is called. For each calculated x value the y value is taken from the appropriate input. The output is calculated at the specified times. This function is considerably faster than “Parametric Interpolation Type 1” on page 2-32.

Note that if the difference between the two x parameters is zero, the output becomes undefined.

2.11.9 Radius



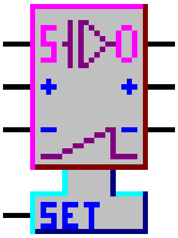
library entry:	FncRadius
length:	30 bytes
inputs:	2 analog inputs
outputs:	2 analog outputs
parameters:	time mask, thick count and norm factor
purpose:	calculate radius and media thickness from velocity and rpm.

This is a special function block. If you reel some kind of media onto a roll and you are able to measure the velocity of the media and the rounds per minute of the roll, this function may be used to calculate the actual radius of the roll and the thickness of the medium. The following calculations are performed:

$$\mathit{radius} = \mathit{velocity} / \mathit{rpm}$$
$$\mathit{thickness} = \mathbf{norm} \times (\Delta\, \mathit{radius}) / \mathit{rpm}$$

The thickness is calculated only at specified intervals. The parameter “thick count” specifies the number of cycles to wait between two such calculations.

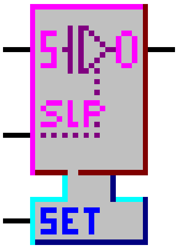
2.11.10 Slope



library entry:	FncSlope1
length:	38 bytes
inputs:	1 analog inputs
	3 digital inputs
outputs:	2 analog outputs
parameters:	time mask, step and two limits
purpose:	Build a slope.

If the “set” input is TRUE, the value from the analog input is put to the analog output. After the “set” input became FALSE, the function starts to work. At each specified cycle the digital inputs are checked. if the input “+” is TRUE, the output is increased by the value of the parameter “step”. If the input “-” is TRUE, the output is decreased by the same value. So if both inputs are TRUE, the output will not change. If the output exceeds either the lower or the upper limit, the respective digital output becomes TRUE.

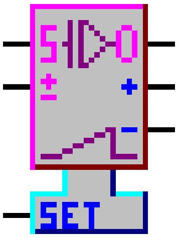
2.11.11 Slope Type 2



library entry:	FncSlp2
length:	20 bytes
inputs:	2 analog inputs
outputs:	1 analog outputs
parameters:	time mask
purpose:	Build a slope.

At the rising edge of the “set” input, the value from the analog input *S* is put to the analog output. At each specified cycle the value from the input *slp* is added to the output.

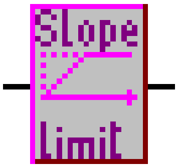
2.11.12 Integrator



library entry:	FncIntegr
length:	36 bytes
inputs:	2 analog inputs 1 digital input
outputs:	1 analog output 2 digital outputs
parameters:	time mask, factor and two limits
purpose:	Build a slope.

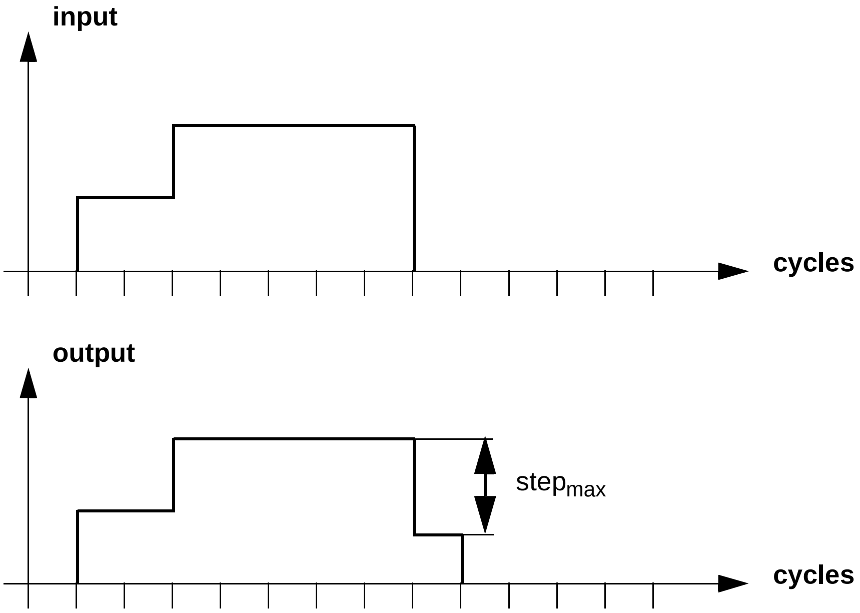
If the “set” input is TRUE, the value from the analog input “S” is put to the analog output. After the “set” input became FALSE, the function starts to work. At each specified cycle the analog input “±” is multiplied by **factor** and added to the output. If the output exceeds either the lower or the upper limit, the respective digital output becomes TRUE.

2.11.13 Limited Slope

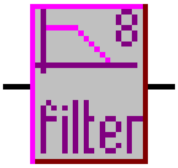


library entry:	FncSlope2
length:	38 bytes
inputs:	1 analog input
outputs:	1 analog output
parameters:	time mask, limit
purpose:	Build a slope.

This function limits the speed of output changes. If the input step is smaller than the parameter “limit”, the output will follow the input. If the input step is larger, the output will be increased or decreased only by the value of “step”. So this function works like a low pass filter.

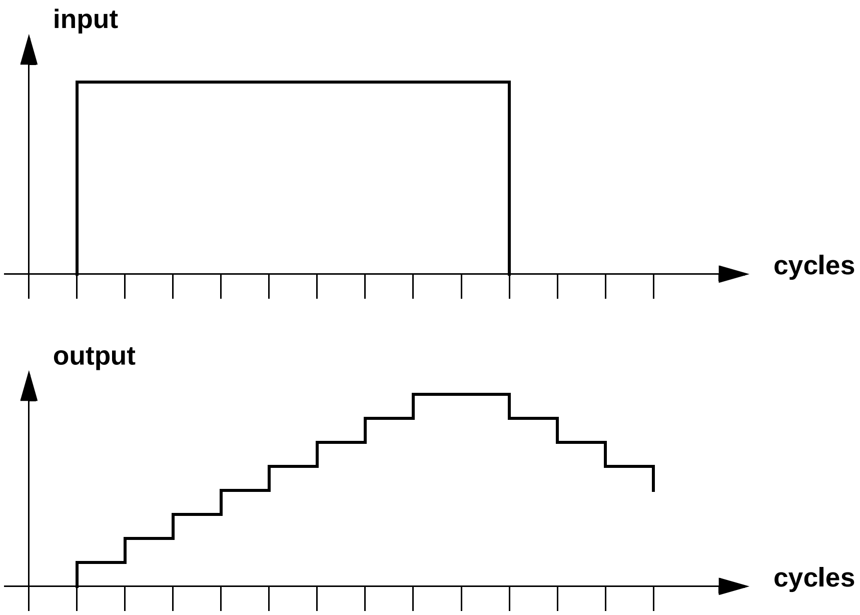


2.11.14 Filter

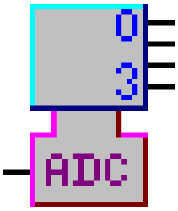


library entry:	FncFilter
length:	48 bytes
inputs:	1 analog input
outputs:	1 analog output
parameters:	time mask
purpose:	Limit edges.

This function limits the speed of output changes.



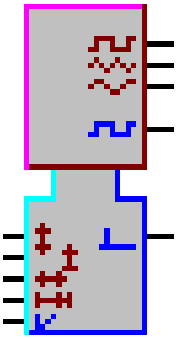
2.11.15 Analog Digital Converter



library entry:	FncAdc
length:	28 bytes
inputs:	1 analog input
outputs:	4 digital outputs
parameters:	time mask, threshold, step
purpose:	Convert analog to digital.

This function converts the analog input into one of 16 digital values. If the analog value is less than “*threshold + step*”, all digital outputs will be 0. If the analog input exceeds “*threshold*” + **n** * “*step*”, the digital outputs are set to represent the binary value **n**.

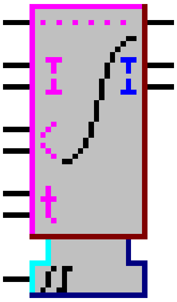
2.11.16 Function Generator



library entry:	FncFGen
length:	68 bytes
inputs:	4 analog inputs 1 digital inputs (enable)
outputs:	3 analog outputs 2 digital outputs
parameters:	amplitude, offset, frequency, duty cycle factors of control inputs time mask
purpose:	generation of signals

This function generates sine, triangle and rectangle signals, respectively. The trigger output becomes one at every start of a cycle. The *enable* input has to be one to calculate this function. The control inputs may be used to change the parameters of the output signals.

2.11.17 Rounding

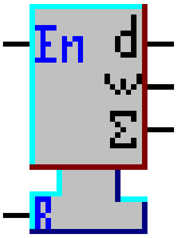


library entry:	FncRound
length:	32 bytes
inputs:	7 analog inputs 1 digital input (enable)
outputs:	1 analog output 2 digital outputs (limit reached)
purpose:	rounding of an input jump

The first input receives the signal which should be rounded. Inputs 2 and 3 are used to limit the output signal - if the output is limited, the appropriate digital output will be set. At inputs 4 and 5 the maximum changing of the output signal per second has to be set for rising and falling edges, respectively. The inputs 6 and 7 get the time (in seconds) for the upper and lower rounding, respectively. These times should not differ too much to assure proper functionality.

If the digital input is set to FALSE, the output value will be the same as the input value; the function is turned off.

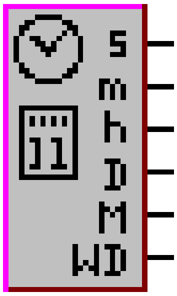
2.11.18 Operating Time Counter



library entry:	FncBCnt
length:	20 bytes
inputs:	2 digital inputs
outputs:	3 analog outputs
purpose:	counting of operating time

The reset input is used to clear the sum counter. The day and week counters are cleared at midnight (the week counter sunday, 0:00) if the user software is running at this time. The resolution of the counter is one minute, the output values are hours.

2.11.19 Calendar



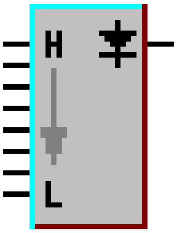
library entry: FncCalendar
length: 18 Bytes
outputs: 6 analog outputs
purpose: output of date and time

At the output *s* the actual second (0..59) is written.
At the output *m* the actual minute (0..59) is written.
At the output *h* the actual hour (0..23) is written.
At the output *D* the actual date (1..31) is written.
At the output *M* the actual month (1..12) is written.
At the output *WD* the actual week day (0..6) is written.

weekday	output
Sunday	0
Monday	1
Tuesday	2
Wednesday	3
Thursday	4
Friday	5
Saturday	6

The values may be used to turn on other function blocks or they may be displayed at the programmable terminal ***S83010***.

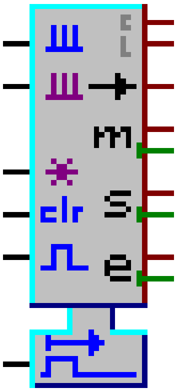
2.11.20 LED-Controller



library entry: FncLED
length: 40 bytes
inputs: 8 digital inputs
outputs: 1 analog outputs
purpose: generation of signals for ***S9704*** Bicolour LEDs

For each connected input one of several kinds of LED states may be selected. If no input is TRUE, the LED remains dark, otherwise it glows or blinks in dependence of the input value with the highest priority. Beside permanent glowing, blinking or pulsating also the colour may be selected. The output of this block is connected directly to the input of the LED output function. If this function block is used for ***S5612*** LEDs, the colours are ignored. On this device the used channel selects one of the four green LEDs followed by three yellow and finally one red LED.

2.11.21 Advanced Analog-/Digital-Counter



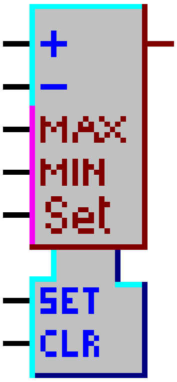
library entry:	FncCntAV
length:	46 bytes
inputs:	2 analog inputs 4 digital inputs
outputs:	9 analog outputs
purpose:	counting of digital pulses or of the difference value of a counter input

At the rising edge of the period input (bottom most) the actual counter value is stored. The output *“s”* contains the start value, the output *“e”* contains the end value of the last period with the corresponding time values (green outputs).

The measurement window input parts a measurement periode in several windows. The pulses are counted for such a window and compared with the maximum value of the actual period. If it is larger than this maximum, it is stored with the current time and becomes the new maximum at the output *“m”*.

The input *“*”* has to contain a multiplication factor for the input. Additionally the output *”c”* contains the current value and the autput *“l”* contains the value of the last window. The counter output is tristate and may be preset. The time outputs may be used to display the time with the function block “Calendar” on page 2-37. The input *clr* is used to slear the value of the current window..

2.11.22 Up-/Down Counter with Limit



library entry:	FncCntUD
length:	24 Bytes
inputs:	3 analog inputs 4 digital inputs
outputs:	1 analog output
purpose:	counting of digital Pulses

If the input *clr* is TRUE, the output is set to zero. If the input *set* is TRUE, the value of the analog input *Set* is put to the output.

If none of these inputs is TRUE, the output is incremented at each rising edge of the input “+”, it is decremented at each rising edge of the input “-”.

If the output becomes larger than the value of the input *max*, it is set to this value. If the output becomes less than the value at the input *min*, it is set to that value.

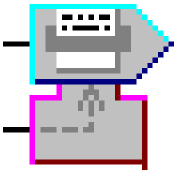
2.11.23 Digital Analog Converter

library entry:	FncDac
length:	10 bytes
inputs:	1 digital input
outputs:	1 analog output
purpose:	Convert digital to analog.

This function converts the digital input into an analog value of either 0 or 1.0.

2.12 Functions for Communication

2.12.1 Message



library entry:	FncMessage
length:	144 bytes
inputs:	1 analog input 1 digital input
parameters:	message, dial string, repeat count, time, modem flag
purpose:	send a message

The ISDN-Option is available only for controllers with built in ISDN.

This function sends a message to the serial port or to the ISDN interface, if available. At the rising edge of the digital input the value from the analog input is appended to the defined message. If “modem flag” is FALSE, this message is put to the serial port. If “modem flag” is TRUE, the “dial- string” is used to dial with the connected modem. If the connection fails, the specified number of retries is done with “time” seconds between. The messages are sent in background so the normal cycles are not disturbed. This function may fail if too many messages were generated and are not yet sent. Therefore a minimum time between two rising edges of the digital input signal should be assured.

This function block may be used to control several displays at the serial interface. The option “formatted message” is used for this purpose. It is possible to show the analog value with a fixed format preceded and/or followed by up to 4 bytes.

If a built in ISDN interface is available, an optional data connection to another ISDN-device is built and the message is sent using X.75 protocol on the B-channel. After sending the message and receiving the acknowledge from the X.75 link, the connection is finished.

It is also possible to send a message to the event log. The message has to be preceded by a unique number followed by a colon, e.g.

123: MESSAGE FROM ABC

The number is not part of the message; it must have a unique value for each used function block sending to the event log.

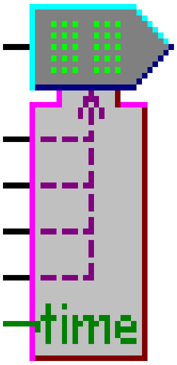
To send a message to the alphameric display of the controller ***S5614*** the option “event log” is used, too. The unique number and the colon are not used in this case. Short messages will be shown static while longer messages will scroll through the display. It is possible to include date and time information using the ‘$’ escape character:

character	example	output	example
Y	$Y	year	97
M	$M	month	02
D	$D	day	25
h	$h	hour	08
m	$m	minute	02
s	$s	second	59
d	$dd	day	Mon
y	$yy	month	Feb

If the last sequence of the message is “$!” the message will be put to the display immediately. Usually system messages like “case open” or “ISDN not connected” have higher priority.

To send a message to the display of a remote ***S5614*** controller, a message beginning with TDIS may be sent using the ISDN option. This message will be shown once. If the message starts with TALA it will be shown permanently until cleared by an empty TDIS message. These messages have higher priority than messages from the user software. The control characters mentioned above are not useable in this case.

2.12.2 Display Control for S5614



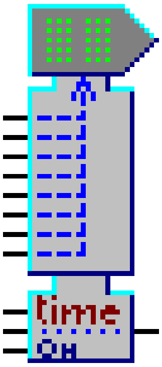
library entry: FncDisp
length: 82 bytes
inputs: 1 digital input
5 analog inputs
purpose: display of analog values, date and time on ***S5614***

Each rising edge of the digital input generates a message to the display.If it contains more than 16 characters, it will be schown running through the display.

A maximum of 4 values may be included into the text. If the input *time* contains a time value from “Calendar” on page 2-37 , this time is used instead of the current system time. To include a value into the text, the sequence **%<N>.<n>f** is used. <N> is the number of digits, <n> is the number of digits after the decimal point. Valid sequences are e.g: **%8.3f %5.0f** etc. These sequences are replaced by the values of the analog inputs. The length of the generated string must not be longer than 250 characters..

See the table above for date and time sequences.

2.12.3 Display of Constant Texts for S5614



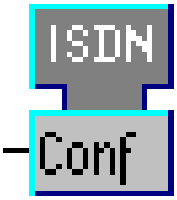
library entry: FncDsptx
length: 168 bytes
inputs: 10 digital inputs
1 analog input
purpose: display of text on ***S5614***

The input *time* indicates the duration for each text in seconds. The enable input *On* is used to activate this function block. If the start input is set, the display of texts starts for all ativated inputs one after the other. After display of the last text is finished the digital output is set to allow starting of another of these function blocks. To each input a text of up to 16 characters may be assigned. If an input is not connected, the previous input may get accordingly more characters assigned.

2.13 ISDN Communication Functions

All the ISDN function blocks are supported only by controllers featuring a built in ISDN interface.

2.13.1 Configuration of the ISDN Interface



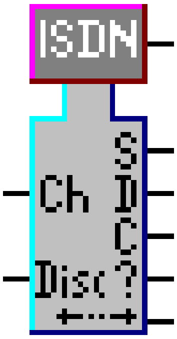
library entry:	FncISDNc
length:	12 bytes
inputs:	1 digital input
parameters:	optional connections, layer 2 behaviour
purpose:	setting of ISDN parameters

It is possible to select if one, two or no optional connections at all are allowed. Optional connections are all connections which are not used to transfer data between function blocks, e.g. configuration calls or speech calls using the ***S9931*** A/B adapter. Incoming and outgoing calls may be allowed or denied separately.

It is also possible to select automatic reactivation of the signalling link. This allows faster building of connections.

Upon reception of a datetime INFO -signal from ISDN, the real time clock may be updated if the difference exceeds one and a half minute.

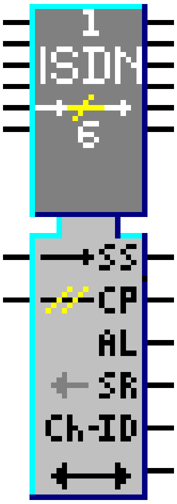
2.13.2 State of an ISDN Channel



library entry:	FncISDNs
length:	24 bytes
inputs:	1 analog input 1 digital input
outputs:	1 analog output 5 digital outputs
parameter:	ISDN channel
purpose:	showing the ISDN state, disconnecting a call

The parameter or the *Ch* input selects one of the B-channels. There are two channels available which may have a connection independently. If both B-channels are in use, an additional incoming call may be detected (knocking). The SETUP message contains information about the kind of the call (data, configuration, speech). The disconnect input may be used to disconnect a call. Only a calls with the corresponding output connected to a wire will be disconnected.

2.13.3 Exchange of Digital Data



library entry:	FncISDNd
length:	120 bytes
inputs:	8 digital inputs
outputs:	1 analog output 11 digital outputs
parameter:	ISDN number, block identification
purpose:	data exchange with a remote controller

Each of two independent systems featuring ISDN uses one of these function blocks. The ISDN number of the remote station and a unique block identification are the parameters to identify the partner. The digital data outputs represent the input values of the remote function block and vice versa. The data transfer may be requested by the external *connect* input or done automatically if one of the input values changes. Upon request or when the data values are stable for some time the transfer is disconnected. The control outputs show the state of the connection.

A data connection will be disconnected if the user software is loaded but it will remain connected if the user program segment containing this function block is disabled.

If the option “pulses” is selected, upon each rising edge of the remote input signal a pulse at the appropriate output line of the function block will be generated for one clock cycle. This option has no influence of the behaviour of the remote function block.

The control output *SS* becomes TRUE after the SETUP message to the remote station is delivered. After reception of CALL PROCEEDING the output *CP* becomes TRUE. After the remote controller has answered the SETUP message with ALERTING or CONNECT the output *AL* will become TRUE. After reception of an incoming call the output *SR* will become TRUE. After building the data link to the remote station the output *connected* is TRUE.

After disconnecting the link all control outputs become FALSE.

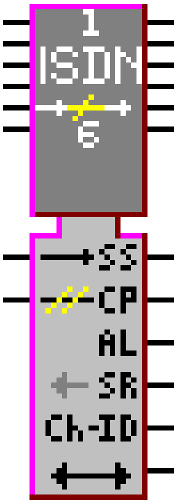
The output *Ch-ID* is -1 if no B channel is used by this function block. Otherwise it will be 0 or 1 and may be used to indicate the occupied B channel to another function block like “State of an ISDN Channel”. The value 2 means reception of data without connection through the “knocking” feature when both channels are used otherwise.

The following parameters are necessary for this function block:

- **ISDN-number:** following the number a comment may be entered. It has to be separated by ‘#’. An asterisk (‘*’) may be used to suppress comparing the following numbers upon reception of a call. This is necessary if the calling MSN is not the same as the receiving MSN.
- **Block identification:** 5 numeric digits which have to be the same at the two function blocks which should exchange data.
- **Automatic setup:** The remote station will be called after a change of an input value is detected.

- **Automatic disconnect:** If the transmitted and the received data do not change for up to 5 minutes the data link will be disconnected.
- **Master/slave:** If both sides initiate a call at the same time the master refuses to accept the incoming call to avoid two connections to the same remote controller at the same time.
- **Alerting before disconnect:** if all input signals are FALSE an incoming call with data TRUE will be refused if the appropriate option of the ISDN configuration is enabled. Sending ALERTING before DISCONNECT allows the remote controller to recognize the reception of the SETUP message.
- **Pulses on output:** After reception of TRUE data signals, the outputs are activated for one cycle only. Otherwise the output signals will be cleared after reception of FALSE data signals.

2.13.4 Exchange of Analog Data



library entry:	FncISDNa
length:	168 bytes
inputs:	6 analog inputs
	2 digital inputs
outputs:	7 analog outputs
	5 digital outputs
parameter:	ISDN number, block identification
purpose:	data exchange with a remote controller

Each of two independent systems featuring ISDN uses one of these function blocks. The ISDN number of the remote station and a unique block identification is the parameter to identify the partner. The digital data outputs represent the input values of the remote function block and vice versa. The data transfer may be requested by the external *connect* input or done automatically if one of the input values changes. Upon request or when the data values are stable for some time the transfer is disconnected. The control outputs show the state of the connection.

A data connection will be disconnected if the user software is loaded but it will remain connected if the user program segment containing this function block is disabled.

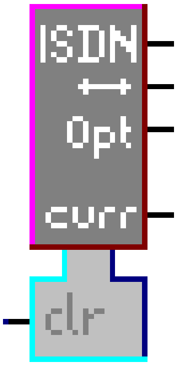
The control output *SS* becomes TRUE after the SETUP message to the remote station is delivered. After reception of CALL PROCEEDING the output *CP* becomes TRUE. When he remote controller has answered the SETUP message with ALERTING or CONNECT the output *AL* will become TRUE. After reception of an incoming call the output *SR* will become TRUE. After building the data link to the remote station the output *connected* is TRUE. After disconnecting the link all control outputs are FALSE.

The output *Ch-ID* is -1 if no B-channel is used by this function block. Otherwise it will be 0 or 1 and may be used to indicate the occupied B channel to another function block like “State of an ISDN Channel”.

The following parameters are necessary for this function block:

- **ISDN-number:** following the number a comment may be entered. It has to be separated by ‘#’. An asterisk (‘*’) may be used to suppress comparing the following numbers upon reception of a call. This is necessary if the calling MSN is not the same as the receiving MSN.
- **Block identification:** 5 numeric digits which have to be the same at the two function blocks which should exchange data.
- **Automatic setup:** The remote station will be called after a change of an input value is detected which is greater than the threshold value.
- **Automatic disconnect:** If the transmitted and the received data do not change for up to 5 minutes the data link will be disconnected.
- **Master/slave:** If both sides initiate a call at the same time the master refuses to accept the incoming call to avoid two connections to the same remote controller at the same time.

2.13.5 Charges of an ISDN Connection



library entry:	FncISDNk
length:	28 bytes
inputs:	1 digital input
outputs:	4 analog outputs
parameter:	ISDN channel, behaviour upon program update
purpose:	showing connection fees

The parameter “channel “selects one or both of the B channels. The outputs show the charges for data connections and optional connections and the sum of both. The fee of the current connection is also shown. This output is zero if no connection is present. The value is in local currency. The digital input *clr* is used to reset the outputs. One of the parameters select if the charges should be cleared after loading of a new user program or if they should be kept.

CHAPTER 3 Programming

The ***SEAL S57901*** Software is the tool for programming, compiling, downloading, debugging and for all operations with buffers.

3.1 Hardware Requirements

To run ***S57901*** software you need at least the following hardware components:

- PC IBM-AT or compatible personal computer with 4MB RAM
- Hard disk
- EGA or VGA video interface card and monitor
- Microsoft compatible mouse
- DOS 3.3 or upwards
- Microsoft Windows 3.0 or upwards (3.1 recommended)

3.2 Getting Started

Creating your own programs for your ***SEAL*** control device is very simple. Start as follows:

- Connect the ***S5701*** box to your PC using a standard male-female cable (NOT crossed).
- Install the software (if not installed), using Microsoft ***SETUP.EXE*** from the Windows Program Manager.
- Start the program ***S57901***

3.3 Programming Manager

After starting the ***SEAL S57901*** software, a small window appears which allows the management of your projects. At first configure your hardware requirements. After that select a program file and start the program editor. Draw your program and save it. Calling the compiler generates the loadable files from your configuration and user program. These may be sent to ***S5701***. After the program is loaded, the debug option allows to watch and set variables of your user program. If you are satisfied with the results, the PC may be disconnected from ***S5701***.

Next, you will find a short description of the menu items of the programming manager.

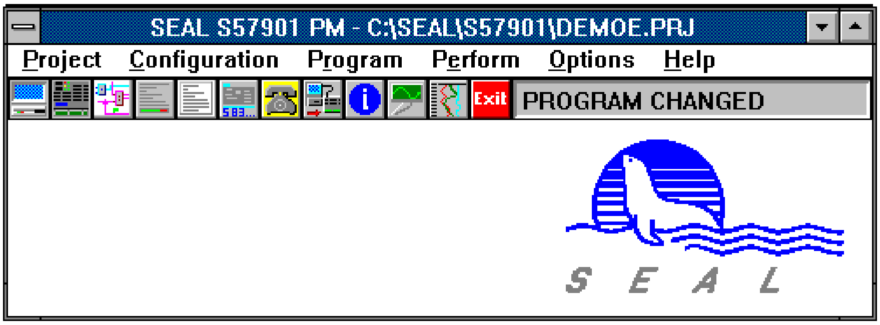

SEAL S57901 PM - C:\SEAL\S57901\DEMOE.PRJ
Project Configuration Program Perform Options Help
S83...
Exit
PROGRAM CHANGED
SEAL

FIGURE 32 Programming Manager Window

3.3.1 Project

- **New**
  Creation of a new project. You will be asked for a configuration and a user program to be used.
- **Open**
  You may select one of the available projects. A project contains a configuration file which describes the hardware and a set of up to 10 program files, one for each segment.
- **Save** or **Save As ...**
  Allows to save the current project.
- **Delete**
  You may delete all dependent files (files generated by the compiler) to save disk space. You may also delete the buffer files or a whole project.
- **Print**
  The print option allows to print the cross reference lists or the I/O configuration if you selected DISPLAY CROSS REFERENCE or DISPLAY I/O REFERENCE from the PERFORM menu before starting the compiler.

3.3.2 Configuration

- **Set**
  A standard file selection menu allows to select a configuration file. Standard config-

urations are supported by the software. Entering a new name allows to create a new configuration. The current configuration may be saved using a new name, too.

- **Edit Configuration**
  In this window you can generate the desired hardware configuration for your own program. In the top left corner of the window the PC is displayed. Double-clicking it with the left mouse button allows to select an I/O port (COM1 or COM2) and a baud rate (usually 9600, 2400 for modem operation). Selecting “Devices”, you choose your hardware and put it into the window pressing the right mouse button. Double-clicking a specific hardware allows to select cycle time and address. The default cycle time is set to 100ms. Selecting a wrong cycle time does not affect the functionality of your software, but selecting time masks in the function configurations will then show wrong entries. In any way the fastest time mask will be the same as the cycle time of the hardware. The selected address of the master device (***S5701***, ***S5703***, ***S5610*** or ***S67001***) has to be 1, the addresses of token bus modules may be any value greater than 5 and must be the same as selected at the dip-switches of the corresponding hardware. Extension modules which are connected to other devices using the serial extension bus must be addressed following the address of the device to which they are connected. After saving this configuration and exiting the configuration editor, this configuration is available for your program files. The program editor needs this configuration to be aware of available in- and outputs.
- **PC Configuration**
  This is a quick way to select port and baud rate of the PC. Use it if you developed the program on another PC and now want to send it to S5701 which is not connected to the same port as it has been at the development PC.
- **S83010 Configuration**
  As a user interface to ***S5701***, the ***SEAL*** industrial terminal ***S83010*** may be used. The graphic LCD display may be configured freely and then down-loaded to the display. This software is described in “Programming of S83010” on page 4-2.

3.3.3 Program

- **Select**
  A standard file selection menu allows to select a program file for each segment. Entering a new name allows to create a new program. The program editor allows to save the current program using a new name. At least one segment must be defined.
- **Delete Segment**
  You may delete a segment from the list. The appropriate program file will not be deleted from the disc. To change the selected program, it is not necessary to delete it previously.
- **Run Conditions**
  For the program as well as for any segment the run conditions may be selected independently. Be aware that a segment will only run when both the run condition of the program and of this segment will be fulfilled. To enter the run conditions, the cron format is used. The default run condition is “always”.
- **Edit Segment**
  On this screen you can generate the desired program for the selected segment. The editor for the user program is started. It is described in “Program Editor” on page 3-4.

3.3.4 Perform

- **Compile**
  The function COMPILE compiles the user program and generates reference files and debugging information, if desired.
- **Transmit**
  The function TRANSMIT loads the translated program, date and time into the connected device (e.g. ***S5701***) and starts the program.
- **Debug**
  This function allows you to monitor selected values or to change values which are not connected to an output clicking the CONST value. All wires which have a debug probe are available. Pressing the left mouse button while on the name of the debug variable, the entry may be moved up or down. Double-clicking the name allows to delete the entry or to insert new entries. All wires which have a name will appear in this selection. Be aware that these changes are lost after recompiling.
- **Buffers**
  The loading, sending and editing of buffers is described in “Buffer Functions” on page 3-7.
- **Display**
  The cross reference lists or the I/O reference as well as the content of a previously loaded buffer may be displayed. During displaying it is possible to print the selected file.

3.3.5 Options

- **Cross Reference List**
  The compiler generates a cross reference list for each segment if this option is selected. In this listing all functions are shown with all wires connected to them.
- **I/O Reference List**
  For each input and output the connected variable and the selected time mask is shown.
- **Auto Save**
  While editing the program file, the file may be saved in specified time intervals.
- **Language**
  You may choose between English and German.

3.4 Program Editor

The graphic editor S57PROG allows to draw a user program like a circuit drawing program. The inputs and outputs as well as several operations are implemented as function blocks. After inserting such a block into the window and moving it to the desired position, connect the pins by simply drawing wires between them.

By selecting **Functions** you choose a function block out of 5 different groups of functions. The functions are described inChapter 2 of this manual. “**I/O functions**” may be selected only after an appropriate device was configured in the **Configuration**. “**Digital Functions**” contain all functions with digital inputs and digital outputs only. “**Analog Functions**” contain all functions with analog inputs and analog outputs only. “**Dig/ana**

Functions” contain the functions containing analog and digital inputs and outputs. “**Buffers**” allow to include buffers for reading or writing variables in different ways. A cyclic buffer will overwrite the oldest entries after being filled whereas a “read once” buffer will stop collecting data after it has been filled. The time mask of the buffer specifies how often the data will be collected. After inserting the buffer symbol you have to connect a digital variable to its enable input. Data collection will only take place if the enable input is HIGH (1). To select the variables which should be collected by the buffer just take the buffer probe symbol out of the buffer or select **Insert Buffer Probe** (this is necessary if the buffer symbol itself is not on the current page or in the current segment). Buffers are independent of segments. Because their enable input must be high to collect data, the segment where the buffer is located must run at least once to set this input to 1.

After placing a function block by pressing and releasing the right mouse button the cursor will display the performed actions after moving the mouse to a pin. Selecting a function by pressing the left mouse button allows to move it around. If the left mouse button is pressed at a pin (the cursor has changed to a soldering device) you can draw a wire to another pin of the same kind. Digital pins may be connected only to digital pins and analog pins may be connected only to analog pins. Outputs may be only connected to inputs if they are not tri-state. Wires which are not connected to an output are displayed in a dotted style. Double clicking a function block shows the wires which are connected to the pins and optionally allows to change parameters or time masks. Parameters are function block specific. Some function blocks (like I/O or PID) support time masks. These are values defining when to calculate these functions. A time mask may be selected out of a list. The fastest time mask is the same as the cycle time of the hardware. During a cycle all functions without a time mask and all functions with the time mask set to the fastest available time mask will be executed. Functions with a time mask set to 2 are executed every other cycle:

cycle:1	2	3	4	5	6	7	8	9	...
time masks: 1	1,2	1,4	1,2	1,8	1,2	1,4	1,2	1,16	...

This assures that during a specific cycle only functions with time masks set to 1 and functions with one of the other time mask will be executed. If you have many time consuming functions which do not have to be executed every cycle you may deliver them to different cycles using different time masks for various groups of functions.

- **Page**
  This menu item allows you to create a new page or to select an existing one. Deleting empty pages is possible, too. The function keys may also be used to switch between pages. F1 to F16 selects page 1 to 16, respectively; SHIFT F1 to SHIFT F16 selects page 10 to 26, respectively. To connect wires between pages, use wire labels. If you want to change the name of the page, double-click the left mouse button in an empty area of the window.
- **View**
  This menu item allows you to change the size of the display. **Home** sets the default size. Larger or smaller sizes reduce system performance.
- **Gridsnap**
  The default value is 4. This gives best drawing results in most cases. If you have dif-

ficulties to select wires, select the whole drawing and use ALIGN_TO_GRID or reduce the GRID SNAP.

- **Edit**
  This menu item allows you to delete or copy selected parts or your program and to paste previous copied or deleted parts. Select several parts on the display using the mouse. Multiple selections are done by pressing the shift key of the keyboard. Now select **Copy** to copy them or **Delete** to delete the selected parts. The DEL key of the keyboard allows to delete selected parts, too. If you want to insert previously deleted or copied parts (only functions and wires are supported) select PASTE. By pressing the left mouse button, the parts will be put to the screen. **Select Net** allows to select a whole wire net. After selecting this menu item, click the left mouse button over the desired wire. All wires connected to this wire will be high lighted, even if you change the page.
- **Perform**
  This menu item allows you to perform several checks and to set **Wire Labels** and **Debug Probes**. Wire labels allow an easy identification of connections in reference listings and during debugging. After placing a wire label, double-clicking it allows to change the name of the wire net. If you want to connect wires from different pages, just give them the same name. This is only possible if the wires are of the same type and if no more than one output is connected to the wire net. All characters are significant. Variables with debug probes may be displayed or changed using the **Debug** option of the S57901 software. Setting a debug probe to a wire which is not connected to an output (a dashed wire) allows to set the value of this wire during debugging. **Include Comment Text** allows to include a line of text into the drawing. If you want to insert more than one line of comment text, just press the SHIFT key of the keyboard during inserting of comment text with the left mouse button. **Checks** show open wire ends or open pins of function blocks, respectively. **Statistics** shows how many functions and wires are used in the current program.
- **Help**
  shows a short description of the available features of the software.
- **File**
  The File menu allows to **Save** or **Print** the current program. **Change Segment** is a quick way to edit the program file of another segment. **Exit** the programming part of the software brings you back to S57901.

3.5 Buffer Functions

3.5.1 File Format

The first line of the file contains the names of the variables and starts with 2 separation characters. Each line of the file contains the date and the time of the data collection and up to 16 entries which represent the data of the variables:

```
1992.08.10 13:15:00.002 1 0 1 0 0 1 3.1415 2.71828 1
1992.08.10 13:15:00.006 1 0 1 0 0 1 3.1415 2.71828 1
1992.08.10 13:15:01.000 1 0 1 1 0 0 8.2 2.70183 0
1992.08.10 13:15:01.003 1 0 1 1 0 0 8.2 2.69422 0
```

This example shows a possible buffer file consisting of 4 entries with 7 digital and 2 analog variables each. For these files the separation character (space, tab or semi-colon) and the type of decimal point (comma or period) may be selected.

3.5.2 Load Buffer

If one or more buffers are defined in a project, their content may be loaded onto the PC. The buffer will be saved to disk in a file named <project>.B<no> where <project> is the name of the project and <no> is the number of the buffer (1... 32). The basic name of the buffer files may be entered by the user. **An already existing file of the same name will be overwritten.**

3.5.3 Display Buffer File

You may select the **File** using a file selection template.

Display Text allows to show a table of the buffer file, **Display Graphics** shows the content in a graphic format; one of the variables is displayed in textformat, too.

Maximizing allows to choose out of several ways to fit the display to the screen.

You may also choose the **Colors** or the **Font**.

3.5.4 Edit Buffer File

The display format is the same as described previously; the entries of date and time are not used here.

New File allows to create new buffer files. You may select the number of variables and the number of entries.

Single Points allows to change each single point of the buffer by pressing the right mouse button at the desired position. Keeping the button pressed allows to draw continous points.

Interpolation is used to set several points of the buffer and afterwards these points may be connected in one of three ways.

You may **Delete** or **Insert** some buffer points after selecting a range with the left mouse button.

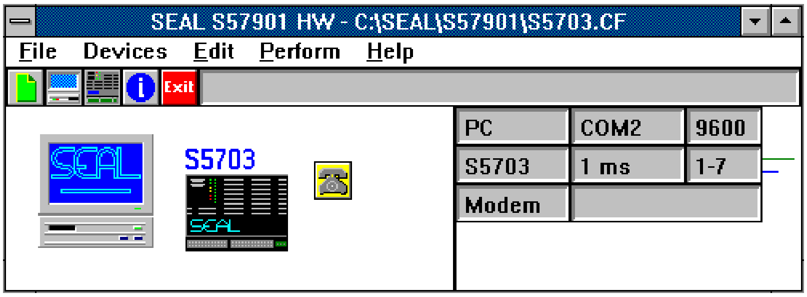
Calculate allows to fill the buffer or parts of it with functions like Sine or Exponent.

3.5.5 Send Buffer

If you have defined writing buffers in your project, you may fill them by sending an appropriate buffer file to the control device. You may create this file by a read buffer, by the software just described or even with another text editor or spreadsheed program.

3.6 An Example Program

Start the ***SEAL S57901*** software. Select PROJECT_NEW. Give a name to the project; select PROJECT_SAVE_AS and enter a name. The next step is to select a configuration. You may select one of the delivered configurations, e.g. ***S5701***. This standard configuration consists of a PC and an ***S5701*** device at address 1 running with 100ms cycle time. Select CONFIGURATION_SET, then enter *myconf* into the file selection, then select CONFIGURATION_EDIT. A window will appear on the screen as shown below, but without any devices. Now insert the hardware devices. Select DEVICE_S5701 and place it somewhere in your window by pressing and releasing the right mouse button. Your screen should now look like the following window.

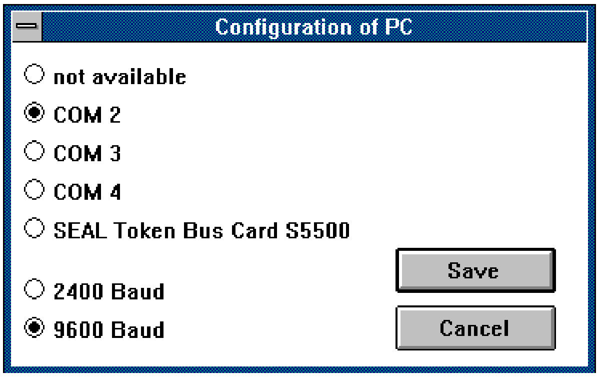


PC	COM2	9600
S5703	1 ms	1-7
Modem		

FIGURE 33 Configuration Window

If you have inserted an ***S5703*** device, several special features are allowed in the menu. If you receive a firmware update (S5703.H), you have to copy it to your ***S57901*** directory. LOAD_FIRMWARE allows to load this new firmware to ***S5703***.

To set the configurations, double-click the PC icon. Now select the communication port (COM1 or COM2) and 9600 baud. Be sure the appropriate dip switch from the ***S5701*** selects 9600 baud, too. Press the SAVE button, then double click the ***S5701*** device. The Device Configuration Template will appear.



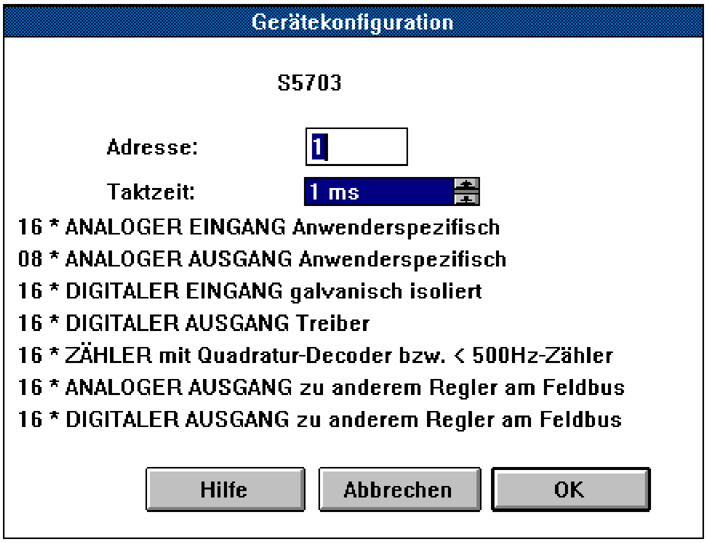
Configuration of PC

○ not available
● COM 2
○ COM 3
○ COM 4
○ SEAL Token Bus Card S5500

○ 2400 Baud
● 9600 Baud

FIGURE 34 PC Configuration

The address of ***S5701*** has to be 1, the cycle time should be selected according to your hardware (100 ms). Press the SAVE button. Now you want to save your hardware configuration by selecting FILE_SAVE and FILE_EXIT. This configuration may be used in future projects.



Gerätekonfiguration

S5703

Adresse: 1

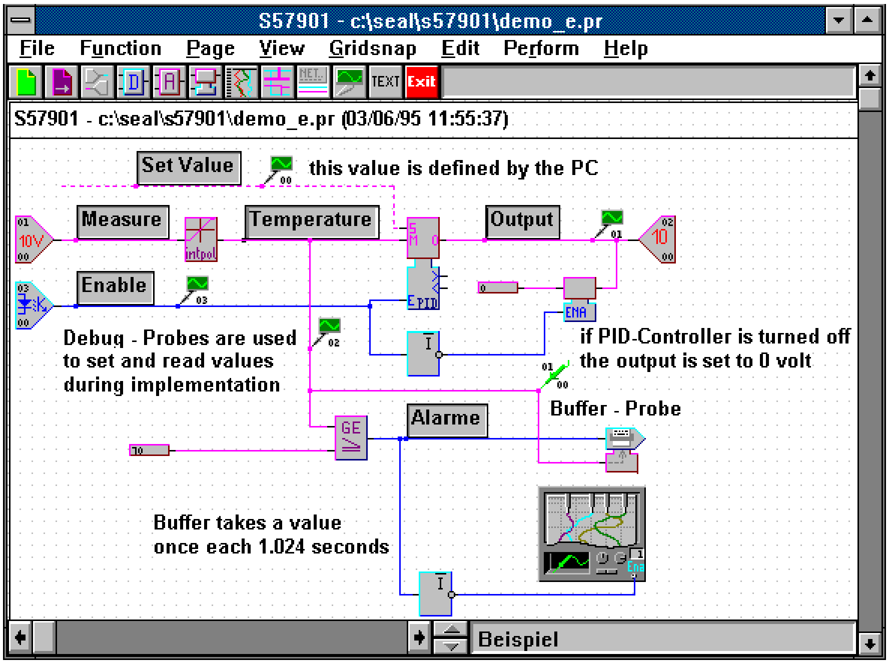
Taktzeit: 1 ms

16 * ANALOGER EINGANG Anwenderspezifisch
08 * ANALOGER AUSGANG Anwenderspezifisch
16 * DIGITALER EINGANG galvanisch isoliert
16 * DIGITALER AUSGANG Treiber
16 * ZÄHLER mit Quadratur-Decoder bzw. < 500Hz-Zähler
16 * ANALOGER AUSGANG zu anderem Regler am Feldbus
16 * DIGITALER AUSGANG zu anderem Regler am Feldbus

FIGURE 35 Configuration of a Device

If you use other hardware components, select their time masks appropriately, too. The field bus addresses have to be selected according to the dip-switch settings.

To edit a sample program, select PROGRAM_SELECT_SEGMENT_1 and enter *myprog* to the selection menu. Now you may edit this new program by selecting PROGRAM_EDIT_SEGMENT_1. An empty window will appear. Now you may enter the example program as shown below.



S57901 - c:\seal\s57901\demo_e.pr (03/06/95 11:55:37)

this value is defined by the PC

Debug - Probes are used to set and read values during implementation

if PID-Controller is turned off the output is set to 0 volt

Buffer takes a value once each 1.024 seconds

FIGURE 36 The Example Program

Select one function after the other and enter as many as you need by clicking the right mouse button. Move the functions by pressing the left mouse button in the middle of the function. A MOVE cursor will appear and the function will be highlighted. If you want to move more than one function, click one after the other while pressing the SHIFT key of the keyboard. You may select a group also using the mouse by pressing the left button in the empty space and then drawing a selection rectangle. If you release the button, all objects within the rectangle will be selected. They may be moved around or deleted by pressing the DEL key or by selecting EDIT_CUT. To copy them, select EDIT_COPY. Previously copied or deleted objects may be placed by selecting EDIT_PASTE and afterwards by clicking the left mouse button.If you have entered all the functions, the next step is to connect the pins. Just move the cursor to a pin. A SOLDERING cursor will appear. Press the mouse button and move into the desired direction. A wire will be drawn. By clicking the left mouse button you may draw it in right angles. To connect the wire to another pin, move to this pin. Again a SOLDERING cursor will appear. By pressing the left mouse button again, the wire will be connected. It is also possible to connect a wire to another wire as long as there is not more than one output connected to it. Only tri-state outputs may be connected together. Functions with enable inputs have tri-state outputs. If you want to keep an open end of a wire, e.g. for debugging purposes, click the right button of the mouse during drawing of a wire. Your program should now look like the example program.

The next step is to enter the parameters of the function blocks. Double-click a function and a configuration template for this function will appear. Some functions don’t have any parameters, then only the wire nets connected to them are shown in this template. After entering the parameters press the SAVE button or the RETURN key. PERFORM_DEBUG_SET_DEBUG_PROBE is now used to set the debug probes onto the wires after giving them a name by selecting PERFORM_SET_WIRE_LABEL. After putting a label to a wire (watch the appearance of the cursor) you have to double click the wire label to enter a name. Now save your program and exit the programming window.

Now you are back to the ***S57901*** programming manager. OPTIONS_CROSS_REFERENCE and OPTIONS_I/O_REFERENCE should be checked. After that select PERFORM_COMPILE. The compiler will compile the program and generate the reference listings which may be displayed. After PERFORM_TRANSMIT the program is loaded to the controller. It may be debugged by entering the DEBUG mode. A list of the variables which have a debug probe will appear on the screen. Enter the values of the variables having a CONST flag like in the example. Digital variables are changed by clicking the CONST entry, analog variables have to be entered after clicking their symbol.

3.7 Time Dependent Starting of Program Segments

The ***SEAL S5701*** device features a battery buffered time keeper chip. You may run your program segments on various different time and date conditions, for example: segment 1 every Monday between 9:00 and 11:00 and segment 2 every Sunday between 17:00 and 17:24 in months January and February. For the whole program and for each segment you may define your run condition. The default run condition is “always”. The run condition for each segment is an ASCII string containing five fields being integer patterns separated by spaces that specify the following:

minute (0-59),
hour (0-23),
day of the month (1-31),
month of the year (1-12),
day of the week (0-6 with 0=Sunday).

Each of these patterns may be either an asterisk (‘*’, meaning all legal values), or a list of elements separated by commas. An element is either a number, or two numbers separated by a minus sign (meaning an inclusive range). Note that the specification of days may be made by two fields (day of the month and day of the week). If both are specified as a list of elements, both are adhered to. For example, 0 0 1,15 * 1 would run a segment on the first and fifteenth of each month, as well as on every Monday. To specify days by only one field, the other field should be set to * (for example, * * * * 1 would run a segment only on Mondays). Both the run condition of the program and of a segment must be fulfilled to execute the segment.

3.8 Files of S57901 Software

***.exe**	the SEAL S57901 Software
s57901.bli	library of devices and functions
s57901.ini	initial file, keeps last project in use
<xxx>.prj	project file; keeps configuration, segments and buffers
<yyy>.pr	program file; one for each used segment
<yyy>.pra	auto save program file; one for each used segment
<xxx>.b<no>	buffer file (no = 1 - 32) of project <xxx>
<xxx>.s83	S83010 Display configuration file
bmlib.bml	Bitmaps for digital variables
bmalib.bml	Bitmaps for analog variables

The following files are generated by the compiler and may be deleted using the DELETE option in the PROJECT menu:

<xxx>.ypr	
<xxx>.zpr	program files for downloading to hardware
<xxx>.yio	i/o configuration for downloading to hardware
<xxx>.bu	buffer configuration for downloading to hardware
<xxx>.io	i/o configuration listing
<yyy>.xr	cross reference listing for segment <yyy>
<yyy>.fnr	function numbers of the segment <yyy>
<xxx>.db	variables displayed during debugging
<xxx>.dba	all variables with names, also for debugging
<xxx>.dbd	values of debug variables without output (set by user)

CHAPTER 4 Display S83010

The ***S83010*** industrial terminal is a display unit predestined for use with ***SEAL S56xx***, ***S57xx***, ***S59xx*** and ***S67001*** devices. The appearance of the display is freely programmable by an easy to use PC software. The basic version supports communication via the serial RS232 interface. Optionally a field bus interface (twisted pair token bus) or an ISDN interface are available.

4.1 Installation

4.1.1 Power Connection

The power supply (9-40V_{DC}) has to be connected to the power connector:

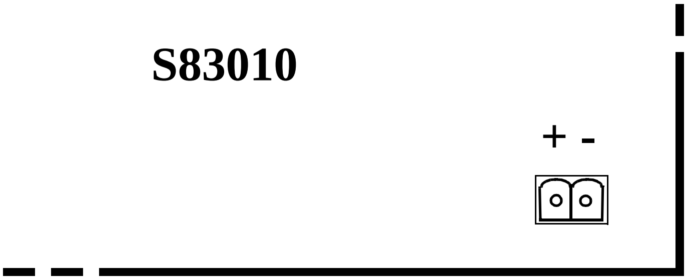


FIGURE 37 S83010: Power Connection

4.1.2 PC and Controller Connection

The basic version of ***S83010*** supports a serial port with 9600 Baud. With a **not crossed** cable (the same you use for the controller) it may be connected to the PC for programming. After the user program has been downloaded to ***S83010***, the display unit has to be connected with a **crossed** RS232 cable to the controller. Use the female 25-pin connector for this purpose.

Pin	RS-232	V.24	Direction	Description
1	AA	101	Both	Protective Ground
2	BA	103	To S83010	Receive Data
3	BB	104	From S83010	Transmit Data
6	CC	107	From S83010	470 Ohm pull up to 12V
7	AB	102	Both	Signal Ground
8	CF	109	From S83010	470 Ohm pull up to 12V

TABLE 25 S83010: RS232 Connector Pin Description

For the connection to the PC or to the controller, only pins 2, 3 and 7 are needed.

4.1.3 LEDs

The three LEDs of ***S83010*** are used to display the status:

- The red LED shows communication errors. Alternatively it may be switched on or off using the user software. Then it shows the value of a digital variable instead of communication errors.
- The yellow LED shows that communication is in progress.
- The green LED shows that the device is running. It may be switched on or off using the user software, too.

4.2 Programming of S83010

Build a user program for one of the ***SEAL*** controllers with the PC software ***S57901***. Variables which should be displayed or set by ***S83010*** must be labeled by applying a wire label to them. Save the program and compile it. The variables are now available to the ***S83010*** configuration software; you may select “S83010 CONFIGURATION”.

A window the same size as the display of ***S83010*** will appear on the screen. Use “EDIT DEFAULT_SIZE” to set the default size if you accidentially resized the window.

You may configure several pages by selecting “NEW PAGE” and switching between these pages in the same way you do in the ***S57901*** program editor. These pages may be selected by the function keys of ***S83010*** if you insert a MENU item on a page.

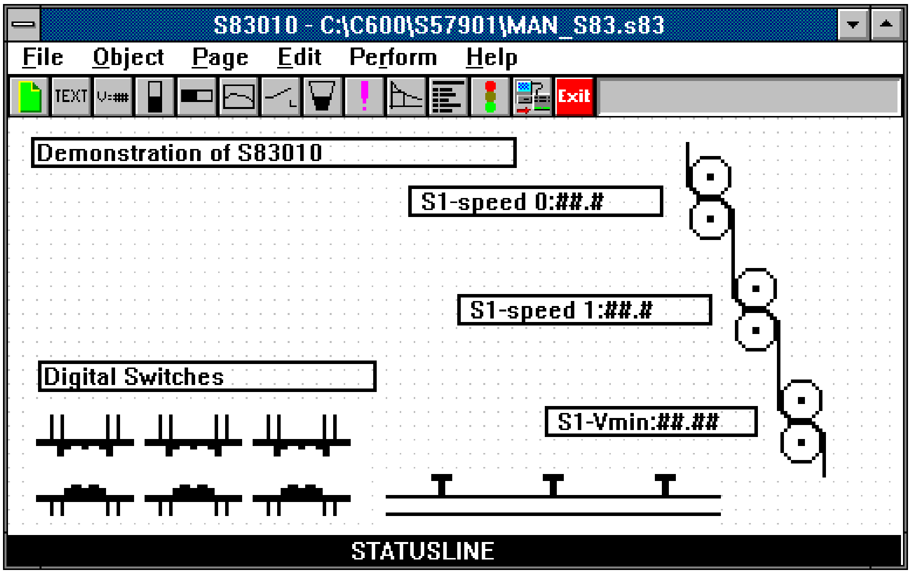


FIGURE 38 Configuration of the display unit S83010

4.2.1 File

- NEW
  You may create a new display configuration file.
- OPEN
  You may select one of the available ***S83010*** configuration files. These files have the extension **S83**. The default configuration file is **<project>.S83**.
- SAVE or SAVE AS
  allows to save the current display configuration.
- EXIT
  returns to ***S57901*** programming manager.

4.2.2 Object

Several kinds of objects may be inserted onto your screen.

- COMMENT TEXT
  allows to insert a line of text. You may select one of five fonts.
- LINE
  Vertical or horizontal lines may be used to build tables. Free lines may be drawn in any angle as needed.
- VARIABLE
  One of the most important objects is a variable. It allows to display or change the value of any named variable of your user program. After inserting a variable, double-click it to open the configuration template. Now you have to select a variable. You also may select a font and whether the name of the variable should be displayed, too. If you select an analog variable, you may choose several floating for-

mats. For digital variables five different display variations are available. If the selected variable is not connected to an output, it may be changed by S83010. The minimum and maximum parameter may be used to set an allowed range. If minimum is the same as maximum, any value may be entered. After saving these parameters, the software calculates the size of the variable object and draws the appropriate frame.

- VERTICAL BAR
- HORIZONTAL BAR
  These bars are graphical representations of analog variables. They may be resized as desired. The minimum and the maximum **must** be defined.
- DIGITAL BITMAP
  A digital variable is displayed as a bitmap. If the value is 0, the left bitmap is used; if the value is 1 (TRUE), the right bitmap of the pair is used. To make user specified bitmaps available, create them as a black/white bitmap of the size 16x16, 16x32, 32x16 or 32x32 bits. Both bitmaps of a pair **must** be of the same size. Then copy them to the file *bmlib.bml*:
  copy bmlib.bml bmlib.ori
  copy bmlib.ori+mybmp0.bmp+mybmp1.bmp bmlib.bml
- ANALOG BITMAP
  For analog bitmaps the first bitmap of the pair of bitmaps shows the outline of the object when it is empty while the second one shows the filled object. The first bitmap will be drawn unconditionally. The second bitmap of the corresponding pair of bitmaps will be drawn from bottom to top dependent of the analog value. If the value is less or equal to the specified minimum, nothing is drawn and the object appears to be empty. If the value reaches the maximum, the whole bitmap will be drawn and the object appears to be filled. The file *bmalib.bml* is used for these bitmaps and may be modified in the same way as described above.
- RECORDER
  The recorder is used to record a variable in suitable time steps. It may be resized as desired. The minimum and the maximum **must** be defined.
- INTERPOLATION CURVE
  This is a special graphical object to display or change interpolation curves. If you use an interpolator with parametric inputs, you may display the interpolation curve with this object. If you keep the inputs open, you may change the curve. You must select the twelve variables for the Y inputs, the input and the output as well as the ranges of the X- and Y axis.
- MENU
  To switch between several user pages, you need menu objects. These allow to assign a page to each of the function keys so, that you may select up to ten other pages from the current one. It is necessary that these pages already exist when you configure the menu.
- LED CONFIGURATION
  is used to turn the red or green LED on or off by the user program. You assign a digital variable to the LED. If this variable is TRUE, the LED will be turned on.
- ALARM
  This is a text message associated to a digital variable. It will be displayed only if the variable is TRUE.

4.2.3 Page

You may create a new page, delete unused pages or switch between the available pages as you do in the program editor.

4.2.4 Edit

You may copy all objects of the current page (if no object is selected) or only selected objects. Afterwards you may insert these objects (on another page) by pasting them. This is an easy way to build several pages with similar layout.

4.2.5 Perform

- TRANSMIT
  is used to send the configuration to the display unit. Connect ***S83010*** to the PC, select “F1 - LOAD FROM PC” from the display’s main menu, and select this command. The configuration will be transmitted. The status line shows the progress. If the transfer has finished, you see the amount of used memory on the display unit. Pressing “ENTER” gets you back to the main menu.

4.2.6 Restricted Usage

If you double click in empty space, you get a password configuration template. This allows to apply a password to the current page. If this page is selected on the display, the user has to enter the password to get access to this page. If he does not enter the right password, the access may be refused or limited to displaying variables without the ability to change them. If the password is “AUTOSTART”, this page will be selected automatically after power on (the first page which owns such a password will be selected).

4.3 Using the Display S83010

After connecting ***S83010*** to the power, it boots from ROM. A total hard reset may be performed if the “RETURN” key is pressed during this time. After the booting has completed, the main menu is displayed. The function keys are used to select menu items. F1 to F5 is selected by pressing the appropriate key. F6 to F10 is entered by pressing SHIFT and the appropriate function key. Red key labels require the SHIFT key, blue labels are used for text input in ALPHA mode and black key labels are the default keys. The HELP key may be pressed to get a help screen. Generally inputs are terminated with RETURN. ESCAPE allows to cancel input.

4.3.1 Main Menu

“System Parameters” allows to select several languages and display options. “Load From PC” starts the downloading function. It times out after some minutes if nothing is received from the PC. “Start User Program” selects the first page of the user program.

4.3.2 User Program

After selecting a user page, the configured page is displayed. If there are objects which allow user input (variables which are not connected to outputs), these objects may be selected using the LEFT or RIGHT cursor key. The currently selected object will be high lighted. If it is a digital variable, the value may be changed by pressing the UP or DOWN cursor key. If the object represents an analog variable, UP or DOWN increments or decrements the value by 1%, respectively. SHIFT UP or SHIFT DOWN changes the value by 10% of the difference between configured minimum and maximum. By pressing the MODE key, a value may be entered.

After selecting an interpolator and pressing the MODE key, LEFT and RIGHT may be used to select a point, to shift or to rotate the curve. UP and DOWN changes it like analog variables. Pressing MODE again allows to enter a value. Pressing ENTER transmits the changed value to the connected controller.

Technical Specifications

S5701:

16 Analog Inputs

Resolution:	13bit
Auto-ranging:	∗1, ∗2, ∗4, ∗8, ∗16, ∗32, ∗64, ∗128
Full scale:	$10V_{DC}$ (40kΩ) for channels 0-11
	$22mA_{DC}$ (421Ω) for channels 12-15
Overload protection:	$60V_{DC}$ continuous for voltage inputs
	$7V_{DC}$ continuous for current inputs

8 Analog Outputs

Resolution:	16bit
Full scale:	$10V_{DC}$ for channels 0-5
	$+20mA_{DC}$ (at 500 Ω max) for channels 6 and 7
Overload protection:	Protected with PTC from short circuit to $20V_{DC}$

8 Digital Inputs

Voltage range:	$0\text{-}32V_{DC}$
Threshold:	$9V_{DC}$
Insulation:	$500V_{DC}$

8 Digital Outputs

Max. Voltage:	380V (DC or peak)
Max. Current:	3A
Insulation:	1000V

2 Counters

have quadrature encoder and are connected to digital inputs.

ChanA1	is connected to digital input 0
ChanB1	is connected to digital input 1
ChanA2	is connected to digital input 2
ChanB2	is connected to digital input 3
Null-Index 1	is connected to digital input 4
Null-Index 2	is connected to digital input 5

Max. Frequency:	1MHz

General Features

Data Storage:	150kB for user program and buffers
Battery-Backup:	10 years
Temperature:	Operating 0-55°C Nonoperating -40 to 75°C
Humidity:	5 - 85%, non-condensing
Supply voltage:	9 - 40V_{DC}
Power consumption:	10W
Max. field-bus length:	100m with shielded 100 Ohm twisted-pair cable
Dimensions:	296x250x130mm
Weight:	1950g
Case:	IP54 or metal

S5703:

16 Analog Inputs

Resolution:	13bit
Full scale:	10V_{DC} (40kΩ) or 22mA_{DC} (421Ω) user configurable
Overload protection:	60V_{DC} continuous for voltage inputs 7V_{DC} continuous for current inputs

8 Analog Outputs

Resolution:	16bit
Full scale:	10V_{DC} or +20mA_{DC} user configurable
Overload protection:	Short circuit to 20V_{DC}

16 Digital Inputs

Voltage range:	0-32V_{DC}
Threshold:	9V_{DC}
Insulation:	500V_{DC}

16 Digital Outputs

Max. Voltage:	50V_{DC}
Max. Current:	100mA
Insulation:	7500V

2 Counters

have quadrature encoder and are connected to digital inputs.

ChanA1	is connected to digital input 0
ChanB1	is connected to digital input 1
ChanA2	is connected to digital input 2
ChanB2	is connected to digital input 3
Null-Index 1	is connected to digital input 4
Null-Index 2	is connected to digital input 5

Max. Frequency:	1MHz

General Features

Data Storage:	512kB for firmware, user program and buffers
Battery-Backup:	10 years
Temperature:	Operating 0-55°C Nonoperating -40 to 75°C
Humidity:	5 - 85%, non-condensing
Supply voltage:	15 - 35V_{DC}
Power consumption:	40W
Max. field-bus length:	100m with shielded 100 Ohm twisted-pair cable
Dimensions:	296x250x55mm
Weight:	1950g
Case:	metal

S5611C, S5611D:

4 Analog Inputs, Configurable: Voltage/Current/Pt100/Pt1000

Resolution:	12bit
Range:	$10V_{DC}$ (40kΩ) for voltage inputs $22mA_{DC}$ (421Ω) for current inputs 125/1250Ω for resistance inputs
Overload protection:	$60V_{DC}$ continous for voltage inputs $7V_{DC}$ continous for current inputs

2 Analog Outputs, Configurable: Voltage/Current

Resolution:	8bit
Range:	$+/-10V_{DC}$ for voltage outputs $+20mA_{DC}$ (at 500 Ω max) for current outputs
Overload protection:	short circuit to $20V_{DC}$

8 Digital Inputs

Voltage-Range:	$0\text{-}32V_{DC}$
Threshold:	$9V_{DC}$
Insulation:	$500V_{DC}$

4 Digital Outputs

Max. Voltage:	$125V_{AC/DC}$
Max. Current:	100mA
Insulation:	7500V

8 Counters

Each digital input may be used as a counter.

Max. Frequency:	500 Hz

General Features

Data storage:	500kB-1.5MB for Program and Buffers
Cycle time:	5-400 ms
Battery-Backup:	2 month, in flash 20 years (without buffer data)
Temperature-range:	usage 0-55°C storage -40 bis 75°C
Humidity:	5 - 85%, not condensing
Power supply:	$12\text{-}24V_{DC}$
Power consumption:	5W
Interfaces:	2 x RS232, Fieldbus
Fieldbus:	RS485, 100kBit/s, polarity independent
Max. Fieldbus-length:	1km with shielded 100 Ohm twisted pair cable
Dimensions:	S5611C:162x104x44mm S5611D:288x144x100mm
Weight:	750g
Case:	S5611C:Metal. S5611D:plastic

S5612:

4 Analog Inputs, Configurable: Voltage/Current/Pt100/Pt1000

Resolution:	12bit
Range:	$10V_{DC}$ (40kΩ) for voltage inputs $22mA_{DC}$ (421Ω) for current inputs 125/1250Ω for resistance inputs
Overload protection:	$60V_{DC}$ continous for voltage inputs $7V_{DC}$ continous for current inputs

4 Analog Outputs, Configurable: Voltage/Current

Resolution:	8bit
Range:	$+/-10V_{DC}$ for voltage outputs $+20mA_{DC}$ (at 500 Ω max) for current outputs
Overload protection:	short circuit to $20V_{DC}$

16 Digital Inputs

Voltage-Range:	$0\text{-}32V_{DC}$
Threshold:	$9V_{DC}$
Insulation:	$500V_{DC}$

4 Digital Outputs

Max. Voltage:	$125V_{AC/DC}$
Max. Current:	100mA
Insulation:	7500V

16 Counters

Each digital input may be used as a counter.

Max. Frequency:	500 Hz

General Features

Data storage:	1MB for Program and Buffers
Cycle time:	5-400 ms
Battery-Backup:	2 month, in flash 20 years (without buffer data)
Temperature-range:	usage 0-55°C storage -40 bis 75°C
Humidity:	5 - 85%, not condensing
Power supply:	$12\text{-}24V_{DC}$
Power consumption:	5W
Interfaces:	2 x RS232, Fieldbus
Fieldbus:	RS485, 100kBit/s, polarity independent
ISDN:	Basic Rate, Q931, 2 B Channels Data transfer to remote controllers or PC
Max. Fieldbus-length:	1km with shielded 100 Ohm twisted pair cable
Dimensions:	330x210x40mm
Weight:	750g
Case:	Metal, plastic cover

S5614:

1 Analog Input, Configurable: Voltage/Current/Pt100/Pt1000

Resolution:	12bit
Range:	$10V_{DC}$ (40kΩ) for voltage inputs $22mA_{DC}$ (421Ω) for current inputs 125/1250Ω for resistance inputs
Overload protection:	$60V_{DC}$ continous for voltage inputs $7V_{DC}$ continous for current inputs

8 Digital Inputs

Voltage-Range:	0-$32V_{DC}$
Threshold:	$9V_{DC}$
Insulation:	$500V_{DC}$

4 Digital Outputs

Max. Voltage:	$125V_{AC/DC}$
Max. Current:	100mA
Insulation:	7500V

8 Counters

Each digital input may be used as a counter.

Max. Frequency:	500 Hz

LED-Display

Type:	16 Characters written by user program

General Features

Data storage:	1MB for Program and Buffers
Cycle time:	5-400 ms
Battery-Backup:	2 month, in flash 20 years (without buffer data)
Temperature-range:	usage 0-55°C storage -40 bis 75°C
Humidity:	5 - 85%, not condensing
Power supply:	12-$24V_{DC}$
Power consumption:	5W
Interfaces:	2 x RS232, Fieldbus
Fieldbus:	RS485, 100kBit/s, polarity independent
ISDN:	Basic Rate, Q931, 2 B Channels Data transfer to remote controllers or PC
Ethernet:	twisted pair, TCP/IP
Max. Fieldbus-length:	1km with shielded 100 Ohm twisted pair cable
Dimensions:	330x210x40mm
Weight:	750g
Case:	Metal, plastic cover

S5651 (Expansion Module for S56xx):

16 Digital Inputs

Voltage range:	0-32V_{DC} or passive
Threshold:	9V_{DC} (if sctive configured)
Insulation:	500V_{DC} (if sctive configured)

General Features like S5611C

S5661 (Expansion Module for S56xx):

8 Digital Outputs

Max. Voltage:	125$V_{AC/DC}$
Max. Current:	100mA
Insulation:	7500V

General Features like S5611C

S5671 (Expansion Module for S56xx):

8 Analog Inputs, configurable Voltage/Current/Pt100

Resolution:	12bit
Range:	+/-10V_{DC} (40kΩ) for voltage inputs
	22mA_{DC} (421Ω) for current inputs
	125/1250Ω for resistance inputs
Overload protection:	60V_{DC} continous for voltage inputs
	7V_{DC} continous for current inputs

General Features like S5611C

S5681 (Expansion Module for S56xx):

8 Analog Outputs, configurable Voltage/Current

Resolution:	12bit
Range:	+/-10V_{DC} for voltage outputs
	+20mA_{DC} (at 500 Ω max) for current outputs
Overload protection:	short circuit to 20V_{DC}

General Features like S5611C

S5911 (Extension for S57xx):

16 Analog Inputs

Resolution:	13bit
Range:	$10V_{DC}$ (40kΩ)
Overload protection:	$40V_{DC}$ continous

General Features

Temperature range:	usage 0-55°C storage -40 bis 75°C
Humidity:	5 - 85%, not condensing
Power supply:	9 - $40V_{DC}$
Power consumption:	10W
Fieldbus:	10MBit/s, Tokenbus
Max. fieldbus length:	100m with shielded 100 Ohm twisted pair cable
Dimensions:	300x100x50mm
Weight:	950g

S5912 (Extension for S57xx):

16 Analog Inputs

Resolution:	13bit
Range:	$22mA_{DC}$ (421Ω)
Overload protection:	$40V_{DC}$ continous

General Features

Temperature range:	usage 0-55°C storage -40 bis 75°C
Humidity:	5 - 85%, not condensing
Power supply:	9 - $40V_{DC}$
Power consumption:	10W
Fieldbus:	10MBit/s, Tokenbus
Max. fieldbus length:	100m with shielded 100 Ohm twisted pair cable
Dimensions:	300x100x50mm
Weight:	950g

S83010:

Data Storage:	100kB for user configuration
Battery-Backup:	10 years
Temperature:	Operating 0-55°C Nonoperating -40 to 75°C
Humidity:	5 - 85%, non-condensing
Supply voltage:	9 - 40V_{DC}
Power consumption:	10W
Max. field-bus length:	100m with shielded 100 Ohm twisted-pair cable
Dimensions:	288x144x100mm
Weight:	750g
Case:	front panel insertion

S5500:

Typ:	AT-bus plug in for IBM-PC
Port-Address-Range:	200H to 7F8H
logical Addresses:	2, 4 or 6
max. no. of cards on bus:	3
Fieldbus:	Token-Bus (with **S57xx**) or 100kb/s RS485Fieldbus (with **S56xx**)
Dimensions:	160x110x16mm

Tecnical data are subject to change without notice!